

DTDLib

0.1

Généré par Doxygen 1.7.2

Wed Apr 20 2011 21 :32 :39



# Table des matières

<b>1</b>	<b>Index des espaces de nommage</b>	<b>1</b>
1.1	Liste des espaces de nommage . . . . .	1
<b>2</b>	<b>Index des classes</b>	<b>3</b>
2.1	Hierarchie des classes . . . . .	3
<b>3</b>	<b>Index des classes</b>	<b>5</b>
3.1	Liste des classes . . . . .	5
<b>4</b>	<b>Index des fichiers</b>	<b>7</b>
4.1	Liste des fichiers . . . . .	7
<b>5</b>	<b>Documentation des espaces de nommage</b>	<b>9</b>
5.1	Référence de l'espace de nommage dtd . . . . .	9
5.1.1	Documentation des définitions de type . . . . .	9
5.1.1.1	AttributesList . . . . .	9
<b>6</b>	<b>Documentation des classes</b>	<b>11</b>
6.1	Référence de la structure dtd : :Choice : :_State . . . . .	11
6.1.1	Description détaillée . . . . .	14
6.1.2	Documentation des constructeurs et destructeur . . . . .	14
6.1.2.1	_State . . . . .	14
6.1.3	Documentation des données membres . . . . .	14
6.1.3.1	nextChoosable . . . . .	14
6.2	Référence de la structure dtd : :QuantifiableContent : :_State . . . . .	14
6.2.1	Description détaillée . . . . .	17
6.2.2	Documentation des constructeurs et destructeur . . . . .	17
6.2.2.1	_State . . . . .	17
6.2.3	Documentation des données membres . . . . .	17
6.2.3.1	endToken . . . . .	17
6.2.3.2	firstToken . . . . .	17
6.2.3.3	nextStep . . . . .	17
6.3	Référence de la structure dtd : :Sequence : :_State . . . . .	17
6.3.1	Description détaillée . . . . .	20
6.3.2	Documentation des constructeurs et destructeur . . . . .	20
6.3.2.1	_State . . . . .	20
6.3.3	Documentation des données membres . . . . .	20
6.3.3.1	nextEmbeddedContent . . . . .	20
6.4	Référence de la classe dtd : :AnyContent . . . . .	20
6.4.1	Description détaillée . . . . .	23

6.4.2	Documentation des constructeurs et destructeur . . . . .	23
6.4.2.1	AnyContent . . . . .	23
6.4.2.2	~AnyContent . . . . .	23
6.4.3	Documentation des fonctions membres . . . . .	23
6.4.3.1	accept . . . . .	23
6.4.3.2	validate . . . . .	23
6.4.3.3	validate . . . . .	23
6.5	Référence de la classe dtd : :Attribute . . . . .	24
6.5.1	Description détaillée . . . . .	24
6.5.2	Documentation des constructeurs et destructeur . . . . .	24
6.5.2.1	Attribute . . . . .	24
6.5.2.2	~Attribute . . . . .	24
6.5.3	Documentation des fonctions membres . . . . .	24
6.5.3.1	name . . . . .	24
6.5.4	Documentation des données membres . . . . .	25
6.5.4.1	_name . . . . .	25
6.6	Référence de la structure dtd : :AttributesComparator . . . . .	25
6.6.1	Description détaillée . . . . .	25
6.6.2	Documentation des fonctions membres . . . . .	25
6.6.2.1	operator() . . . . .	25
6.7	Référence de la classe dtd : :BrowsableContent . . . . .	25
6.7.1	Description détaillée . . . . .	28
6.7.2	Documentation des constructeurs et destructeur . . . . .	28
6.7.2.1	BrowsableContent . . . . .	28
6.7.2.2	~BrowsableContent . . . . .	28
6.7.3	Documentation des fonctions membres . . . . .	28
6.7.3.1	_afterValidation . . . . .	28
6.7.3.2	_beforeValidation . . . . .	29
6.7.3.3	_browseDown . . . . .	29
6.7.3.4	_browseUp . . . . .	29
6.7.3.5	_continueValidation . . . . .	30
6.7.3.6	_newValidation . . . . .	30
6.7.3.7	_startValidation . . . . .	30
6.7.3.8	validate . . . . .	30
6.7.3.9	validate . . . . .	30
6.8	Référence de la classe dtd : :Choice . . . . .	30
6.8.1	Description détaillée . . . . .	33
6.8.2	Documentation des définitions de type membres . . . . .	34
6.8.2.1	_ChoosableSet . . . . .	34
6.8.2.2	_StatesStack . . . . .	34
6.8.2.3	ChoosableSet . . . . .	34
6.8.2.4	const_iterator . . . . .	34
6.8.3	Documentation des constructeurs et destructeur . . . . .	34
6.8.3.1	Choice . . . . .	34
6.8.3.2	~Choice . . . . .	34
6.8.4	Documentation des fonctions membres . . . . .	34
6.8.4.1	_afterValidation . . . . .	34
6.8.4.2	_beforeValidation . . . . .	34
6.8.4.3	_continueValidation . . . . .	35
6.8.4.4	_startValidation . . . . .	35

6.8.4.5	accept	35
6.8.4.6	begin	35
6.8.4.7	end	35
6.8.4.8	validate	36
6.8.5	Documentation des données membres	36
6.8.5.1	_choosable	36
6.8.5.2	_stack	36
6.9	Référence de la classe dtd : :Content	36
6.9.1	Description détaillée	38
6.9.2	Documentation des constructeurs et destructeur	38
6.9.2.1	Content	38
6.9.2.2	~Content	38
6.9.3	Documentation des fonctions membres	38
6.9.3.1	accept	38
6.9.3.2	validate	38
6.9.3.3	validate	39
6.10	Référence de la classe dtd : :DTD	39
6.10.1	Description détaillée	40
6.10.2	Documentation des définitions de type membres	40
6.10.2.1	_AttributesLists	40
6.10.2.2	_ElementId	40
6.10.2.3	_Elements	40
6.10.3	Documentation des constructeurs et destructeur	40
6.10.3.1	DTD	40
6.10.3.2	~DTD	40
6.10.4	Documentation des fonctions membres	40
6.10.4.1	_isValid	40
6.10.4.2	accept	40
6.10.4.3	addAttributesList	41
6.10.4.4	addElement	41
6.10.4.5	checkAttributes	41
6.10.4.6	getAttributesList	41
6.10.4.7	getElement	42
6.10.4.8	isValid	42
6.10.4.9	visit	42
6.10.4.10	visit	42
6.10.4.11	visit	42
6.10.5	Documentation des données membres	42
6.10.5.1	_attributesLists	42
6.10.5.2	_elements	42
6.10.5.3	_lastNodesValid	42
6.10.5.4	_validatingRoot	43
6.10.5.5	_validRootName	43
6.11	Référence de la classe dtd : :ElementContent	43
6.11.1	Description détaillée	46
6.11.2	Documentation des constructeurs et destructeur	46
6.11.2.1	ElementContent	46
6.11.2.2	~ElementContent	46
6.12	Référence de la classe dtd : :ElementReference	46
6.12.1	Description détaillée	49

6.12.2	Documentation des constructeurs et destructeur	49
6.12.2.1	ElementReference	49
6.12.2.2	~ElementReference	49
6.12.3	Documentation des fonctions membres	49
6.12.3.1	_startValidation	49
6.12.3.2	accept	50
6.12.3.3	matches	50
6.12.3.4	name	50
6.12.3.5	ns	50
6.12.3.6	visit	50
6.12.3.7	visit	50
6.12.3.8	visit	50
6.12.4	Documentation des données membres	50
6.12.4.1	_dtd	50
6.12.4.2	_matchResult	51
6.12.4.3	_name	51
6.12.4.4	_namespace	51
6.13	Référence de la classe dtd : :EmptyContent	51
6.13.1	Description détaillée	54
6.13.2	Documentation des constructeurs et destructeur	54
6.13.2.1	EmptyContent	54
6.13.2.2	~EmptyContent	54
6.13.3	Documentation des fonctions membres	54
6.13.3.1	accept	54
6.13.3.2	validate	54
6.13.3.3	validate	54
6.14	Référence de la classe dtd : :InterfaceDTDVisitor	55
6.14.1	Description détaillée	57
6.14.2	Documentation des constructeurs et destructeur	57
6.14.2.1	~InterfaceDTDVisitor	57
6.14.3	Documentation des fonctions membres	57
6.14.3.1	visit	57
6.14.3.2	visit	58
6.14.3.3	visit	58
6.14.3.4	visit	58
6.14.3.5	visit	59
6.14.3.6	visit	59
6.14.3.7	visit	59
6.14.3.8	visit	59
6.14.3.9	visit	59
6.14.3.10	visit	59
6.14.3.11	visitAttributesList	59
6.14.3.12	visitElement	60
6.14.4	Documentation des fonctions amies et associées	60
6.14.4.1	AnyContent	60
6.14.4.2	Choice	60
6.14.4.3	ElementReference	60
6.14.4.4	EmptyContent	60
6.14.4.5	MixedContent	60
6.14.4.6	OptionalContent	60

6.14.4.7	RepeatableContent	60
6.14.4.8	RepeatedContent	61
6.14.4.9	Sequence	61
6.14.4.10	TextContent	61
6.15	Référence de la classe dtd : :MixedContent	61
6.15.1	Description détaillée	64
6.15.2	Documentation des définitions de type membres	64
6.15.2.1	_StatesStack	64
6.15.2.2	ChoosableSet	65
6.15.2.3	const_iterator	65
6.15.3	Documentation des constructeurs et destructeur	65
6.15.3.1	MixedContent	65
6.15.3.2	~MixedContent	65
6.15.4	Documentation des fonctions membres	65
6.15.4.1	_afterValidation	65
6.15.4.2	_beforeValidation	65
6.15.4.3	_continueValidation	65
6.15.4.4	_startValidation	66
6.15.4.5	accept	66
6.15.4.6	begin	66
6.15.4.7	end	66
6.15.4.8	validate	66
6.15.5	Documentation des données membres	67
6.15.5.1	_choice	67
6.15.5.2	_stack	67
6.15.5.3	_textContent	67
6.16	Référence de la classe dtd : :OptionalContent	67
6.16.1	Description détaillée	70
6.16.2	Documentation des constructeurs et destructeur	70
6.16.2.1	OptionalContent	70
6.16.2.2	~OptionalContent	70
6.16.3	Documentation des fonctions membres	70
6.16.3.1	_continueValidation	70
6.16.3.2	_startValidation	71
6.16.3.3	accept	71
6.17	Référence de la classe dtd : :OutputDTDVisitor	71
6.17.1	Description détaillée	75
6.17.2	Documentation des constructeurs et destructeur	75
6.17.2.1	OutputDTDVisitor	75
6.17.2.2	~OutputDTDVisitor	75
6.17.3	Documentation des fonctions membres	75
6.17.3.1	doIndent	75
6.17.3.2	visit	75
6.17.3.3	visit	75
6.17.3.4	visit	75
6.17.3.5	visit	76
6.17.3.6	visit	76
6.17.3.7	visit	76
6.17.3.8	visit	76
6.17.3.9	visit	76

6.17.3.10	visit	76
6.17.3.11	visit	76
6.17.3.12	visitAttributesList	77
6.17.3.13	visitElement	77
6.17.4	Documentation des données membres	77
6.17.4.1	_indent	77
6.17.4.2	_indentChar	77
6.17.4.3	_indentUnit	77
6.17.4.4	_out	77
6.17.4.5	ANY_CONTENT_STR	77
6.17.4.6	ATTLIST_MARKUP_NAME_STR	78
6.17.4.7	CHOICE_SEPARATOR_STR	78
6.17.4.8	CLOSE_MARKUP_STR	78
6.17.4.9	DEFAULT_INDENT_CHAR	78
6.17.4.10	DEFAULT_INDENT_UNIT	78
6.17.4.11	ELEMENT_MARKUP_NAME_STR	78
6.17.4.12	EMPTY_CONTENT_STR	78
6.17.4.13	END_CHOICE_STR	78
6.17.4.14	END_MIXED_STR	78
6.17.4.15	END_SEQUENCE_STR	79
6.17.4.16	INSIDE_MARKUP_SPACE_STR	79
6.17.4.17	MIXED_SEPARATOR_STR	79
6.17.4.18	OPEN_MARKUP_STR	79
6.17.4.19	OPTIONAL_QUANTIFIER_STR	79
6.17.4.20	REPEATABLE_QUANTIFIER_STR	79
6.17.4.21	REPEATED_QUANTIFIER_STR	79
6.17.4.22	SEQUENCE_SEPARATOR_STR	79
6.17.4.23	START_CHOICE_STR	79
6.17.4.24	START_MIXED_STR	80
6.17.4.25	START_SEQUENCE_STR	80
6.17.4.26	TEXT_CONTENT_STR	80
6.18	Référence de la classe dtd : :QuantifiableContent	80
6.18.1	Description détaillée	83
6.18.2	Documentation des constructeurs et destructeur	83
6.18.2.1	QuantifiableContent	83
6.18.2.2	~QuantifiableContent	83
6.19	Référence de la classe dtd : :QuantifiedContent	83
6.19.1	Description détaillée	86
6.19.2	Documentation des définitions de type membres	86
6.19.2.1	_StatesStack	86
6.19.3	Documentation des constructeurs et destructeur	86
6.19.3.1	QuantifiedContent	86
6.19.3.2	~QuantifiedContent	86
6.19.4	Documentation des fonctions membres	87
6.19.4.1	_afterValidation	87
6.19.4.2	_beforeValidation	87
6.19.4.3	content	87
6.19.4.4	content	87
6.19.4.5	validate	87
6.19.5	Documentation des données membres	87



6.19.5.1	_content	87
6.19.5.2	_stack	87
6.20	Référence de la classe dtd : :RepeatabContent	88
6.20.1	Description détaillée	91
6.20.2	Documentation des constructeurs et destructeur	91
6.20.2.1	RepeatabContent	91
6.20.2.2	~RepeatabContent	91
6.20.3	Documentation des fonctions membres	91
6.20.3.1	_continueValidation	91
6.20.3.2	_startValidation	92
6.20.3.3	accept	92
6.21	Référence de la classe dtd : :RepeatedContent	92
6.21.1	Description détaillée	95
6.21.2	Documentation des constructeurs et destructeur	95
6.21.2.1	RepeatedContent	95
6.21.2.2	~RepeatedContent	95
6.21.3	Documentation des fonctions membres	95
6.21.3.1	_continueValidation	95
6.21.3.2	_startValidation	96
6.21.3.3	accept	96
6.22	Référence de la classe dtd : :Sequence	96
6.22.1	Description détaillée	99
6.22.2	Documentation des définitions de type membres	100
6.22.2.1	_OrderedContent	100
6.22.2.2	_StatesStack	100
6.22.2.3	const_iterator	100
6.22.2.4	OrderedContent	100
6.22.3	Documentation des constructeurs et destructeur	100
6.22.3.1	Sequence	100
6.22.3.2	~Sequence	100
6.22.4	Documentation des fonctions membres	100
6.22.4.1	_afterValidation	100
6.22.4.2	_beforeValidation	100
6.22.4.3	_continueValidation	101
6.22.4.4	_startValidation	101
6.22.4.5	accept	101
6.22.4.6	begin	101
6.22.4.7	end	102
6.22.4.8	validate	102
6.22.5	Documentation des données membres	102
6.22.5.1	_embeddedContent	102
6.22.5.2	_stack	102
6.23	Référence de la classe dtd : :TextContent	102
6.23.1	Description détaillée	105
6.23.2	Documentation des constructeurs et destructeur	105
6.23.2.1	TextContent	105
6.23.2.2	~TextContent	105
6.23.3	Documentation des fonctions membres	105
6.23.3.1	_startValidation	105
6.23.3.2	accept	105

6.23.3.3	visit	106
6.23.3.4	visit	106
6.23.3.5	visit	106
6.23.4	Documentation des données membres	106
6.23.4.1	_validationResult	106
<b>7</b>	<b>Documentation des fichiers</b>	<b>107</b>
7.1	Référence du fichier src/AnyContent.cpp	107
7.2	Référence du fichier src/AnyContent.hh	107
7.3	Référence du fichier src/Attribute.cpp	108
7.4	Référence du fichier src/Attribute.hh	109
7.5	Référence du fichier src/AttributesList.hh	110
7.6	Référence du fichier src/BrowsableContent.cpp	112
7.7	Référence du fichier src/BrowsableContent.hh	112
7.8	Référence du fichier src/Choice.cpp	113
7.9	Référence du fichier src/Choice.hh	114
7.10	Référence du fichier src/Content.cpp	116
7.11	Référence du fichier src/Content.hh	116
7.12	Référence du fichier src/DTD.cpp	117
7.13	Référence du fichier src/DTD.hh	118
7.14	Référence du fichier src/ElementContent.cpp	119
7.15	Référence du fichier src/ElementContent.hh	120
7.16	Référence du fichier src/ElementReference.cpp	122
7.17	Référence du fichier src/ElementReference.hh	122
7.18	Référence du fichier src/EmptyContent.cpp	124
7.19	Référence du fichier src/EmptyContent.hh	124
7.20	Référence du fichier src/InterfaceDTDVisitor.hpp	125
7.21	Référence du fichier src/MixedContent.cpp	126
7.22	Référence du fichier src/MixedContent.hh	127
7.23	Référence du fichier src/OptionalContent.cpp	129
7.24	Référence du fichier src/OptionalContent.hh	130
7.25	Référence du fichier src/OutputDTDVisitor.cpp	131
7.26	Référence du fichier src/OutputDTDVisitor.hh	132
7.27	Référence du fichier src/QuantifiableContent.cpp	134
7.28	Référence du fichier src/QuantifiableContent.hh	134
7.29	Référence du fichier src/QuantifiedContent.cpp	135
7.30	Référence du fichier src/QuantifiedContent.hh	136
7.31	Référence du fichier src/RepeatableContent.cpp	138
7.32	Référence du fichier src/RepeatableContent.hh	138
7.33	Référence du fichier src/RepeatedContent.cpp	140
7.34	Référence du fichier src/RepeatedContent.hh	141
7.35	Référence du fichier src/Sequence.cpp	142
7.36	Référence du fichier src/Sequence.hh	143
7.37	Référence du fichier src/TextContent.cpp	145
7.38	Référence du fichier src/TextContent.hh	145

# Chapitre 1

## Index des espaces de nommage

### 1.1 Liste des espaces de nommage

Liste de tous les espaces de nommage avec une brève description :

<a href="#">dtd</a> . . . . .	9
-------------------------------	---



## Chapitre 2

# Index des classes

### 2.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

dtd : :QuantifiableContent : :_State . . . . .	14
dtd : :Choice : :_State . . . . .	11
dtd : :Sequence : :_State . . . . .	17
dtd : :Attribute . . . . .	24
dtd : :AttributesComparator . . . . .	25
dtd : :Content . . . . .	36
dtd : :AnyContent . . . . .	20
dtd : :BrowsableContent . . . . .	25
dtd : :QuantifiableContent . . . . .	80
dtd : :ElementContent . . . . .	43
dtd : :Choice . . . . .	30
dtd : :ElementReference . . . . .	46
dtd : :QuantifiedContent . . . . .	83
dtd : :OptionalContent . . . . .	67
dtd : :RepeatableContent . . . . .	88
dtd : :RepeatedContent . . . . .	92
dtd : :Sequence . . . . .	96
dtd : :MixedContent . . . . .	61
dtd : :TextContent . . . . .	102
dtd : :EmptyContent . . . . .	51
dtd : :DTD . . . . .	39
dtd : :InterfaceDTDVisitor . . . . .	55
dtd : :OutputDTDVisitor . . . . .	71



## Chapitre 3

# Index des classes

### 3.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

<a href="#">dtd : :Choice : :_State</a>	11
<a href="#">dtd : :QuantifiableContent : :_State</a>	14
<a href="#">dtd : :Sequence : :_State</a>	17
<a href="#">dtd : :AnyContent</a>	20
<a href="#">dtd : :Attribute</a>	24
<a href="#">dtd : :AttributesComparator</a>	25
<a href="#">dtd : :BrowsableContent</a>	25
<a href="#">dtd : :Choice</a>	30
<a href="#">dtd : :Content</a>	36
<a href="#">dtd : :DTD</a>	39
<a href="#">dtd : :ElementContent</a>	43
<a href="#">dtd : :ElementReference</a>	46
<a href="#">dtd : :EmptyContent</a>	51
<a href="#">dtd : :InterfaceDTDVisitor</a>	55
<a href="#">dtd : :MixedContent</a>	61
<a href="#">dtd : :OptionalContent</a>	67
<a href="#">dtd : :OutputDTDVisitor</a>	71
<a href="#">dtd : :QuantifiableContent</a>	80
<a href="#">dtd : :QuantifiedContent</a>	83
<a href="#">dtd : :RepeatableContent</a>	88
<a href="#">dtd : :RepeatedContent</a>	92
<a href="#">dtd : :Sequence</a>	96
<a href="#">dtd : :TextContent</a>	102





## Chapitre 4

# Index des fichiers

### 4.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

src/ <a href="#">AnyContent.cpp</a>	107
src/ <a href="#">AnyContent.hh</a>	107
src/ <a href="#">Attribute.cpp</a>	108
src/ <a href="#">Attribute.hh</a>	109
src/ <a href="#">AttributesList.hh</a>	110
src/ <a href="#">BrowsableContent.cpp</a>	112
src/ <a href="#">BrowsableContent.hh</a>	112
src/ <a href="#">Choice.cpp</a>	113
src/ <a href="#">Choice.hh</a>	114
src/ <a href="#">Content.cpp</a>	116
src/ <a href="#">Content.hh</a>	116
src/ <a href="#">DTD.cpp</a>	117
src/ <a href="#">DTD.hh</a>	118
src/ <a href="#">ElementContent.cpp</a>	119
src/ <a href="#">ElementContent.hh</a>	120
src/ <a href="#">ElementReference.cpp</a>	122
src/ <a href="#">ElementReference.hh</a>	122
src/ <a href="#">EmptyContent.cpp</a>	124
src/ <a href="#">EmptyContent.hh</a>	124
src/ <a href="#">InterfaceDTDVisitor.hpp</a>	125
src/ <a href="#">MixedContent.cpp</a>	126
src/ <a href="#">MixedContent.hh</a>	127
src/ <a href="#">OptionalContent.cpp</a>	129
src/ <a href="#">OptionalContent.hh</a>	130
src/ <a href="#">OutputDTDVisitor.cpp</a>	131
src/ <a href="#">OutputDTDVisitor.hh</a>	132
src/ <a href="#">QuantifiableContent.cpp</a>	134
src/ <a href="#">QuantifiableContent.hh</a>	134
src/ <a href="#">QuantifiedContent.cpp</a>	135

src/QuantifiedContent.hh	136
src/RepeatableContent.cpp	138
src/RepeatableContent.hh	138
src/RepeatedContent.cpp	140
src/RepeatedContent.hh	141
src/Sequence.cpp	142
src/Sequence.hh	143
src/TextContent.cpp	145
src/TextContent.hh	145

## Chapitre 5

# Documentation des espaces de nommage

### 5.1 Référence de l'espace de nommage dtd

#### Classes

- class [AnyContent](#)
- class [Attribute](#)
- struct [AttributesComparator](#)
- class [BrowsableContent](#)
- class [Choice](#)
- class [Content](#)
- class [DTD](#)
- class [ElementContent](#)
- class [ElementReference](#)
- class [EmptyContent](#)
- class [InterfaceDTDVisitor](#)
- class [MixedContent](#)
- class [OptionalContent](#)
- class [OutputDTDVisitor](#)
- class [QuantifiableContent](#)
- class [QuantifiedContent](#)
- class [RepeatableContent](#)
- class [RepeatedContent](#)
- class [Sequence](#)
- class [TextContent](#)

#### Définition de type

- typedef std : :set< [Attribute](#) \*, [AttributesComparator](#) > [AttributesList](#)

#### 5.1.1 Documentation des définitions de type

- 5.1.1.1 typedef std : :set<[Attribute](#)\*, [AttributesComparator](#)> dtd : :[AttributesList](#)

Définition à la ligne 28 du fichier [AttributesList.hh](#).



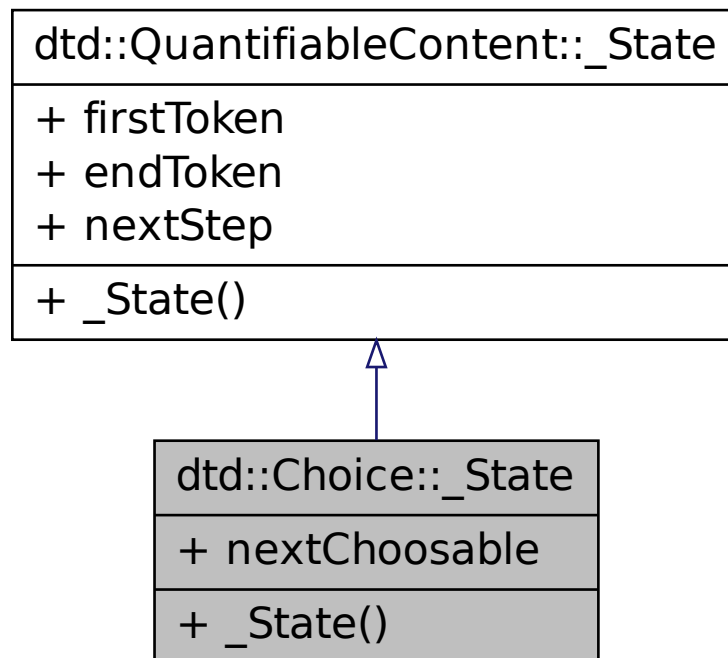
## Chapitre 6

# Documentation des classes

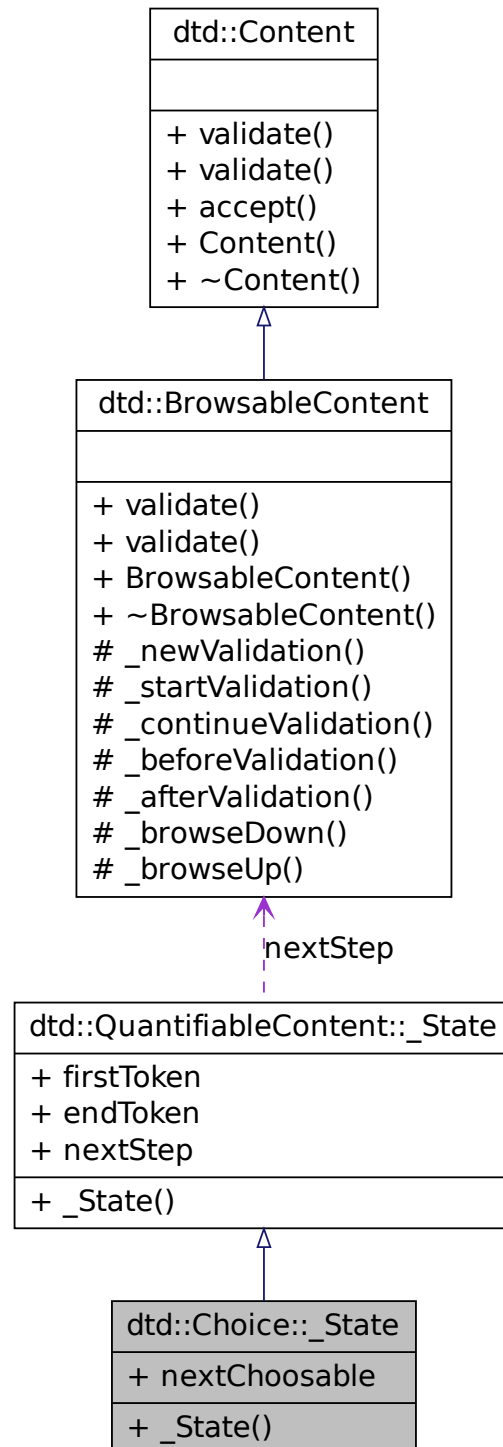
### 6.1 Référence de la structure dtd : :Choice : :\_State

```
#include <Choice.hh>
```

Graphe d'héritage de dtd : :Choice : :\_State :



Graphe de collaboration de dtd : :Choice : :\_State :



## Fonctions membres publiques

- [\\_State](#) (xml : :CompositeMarkupNode : :ChildrenIterator aFirstToken, xml : :CompositeMarkupNode : :ChildrenIterator anEndToken, [BrowsableContent](#) \*aNextStep, [\\_ChoosableSet](#) : :iterator aNextChoosable)

## Attributs publics

- [\\_ChoosableSet](#) : :iterator [nextChoosable](#)

### 6.1.1 Description détaillée

Définition à la ligne 78 du fichier Choice.hh.

### 6.1.2 Documentation des constructeurs et destructeur

- 6.1.2.1 **dtd : :Choice : :\_State ( xml : :CompositeMarkupNode : :ChildrenIterator *aFirstToken*, xml : :CompositeMarkupNode : :ChildrenIterator *anEndToken*, BrowsableContent \* *aNextStep*, [\\_ChoosableSet](#) : :iterator *aNextChoosable* )**  
[inline]

Définition à la ligne 82 du fichier Choice.hh.

### 6.1.3 Documentation des données membres

- 6.1.3.1 **[\\_ChoosableSet](#) : :iterator dtd : :Choice : :\_State : :nextChoosable**

Définition à la ligne 80 du fichier Choice.hh.

La documentation de cette structure a été générée à partir du fichier suivant :

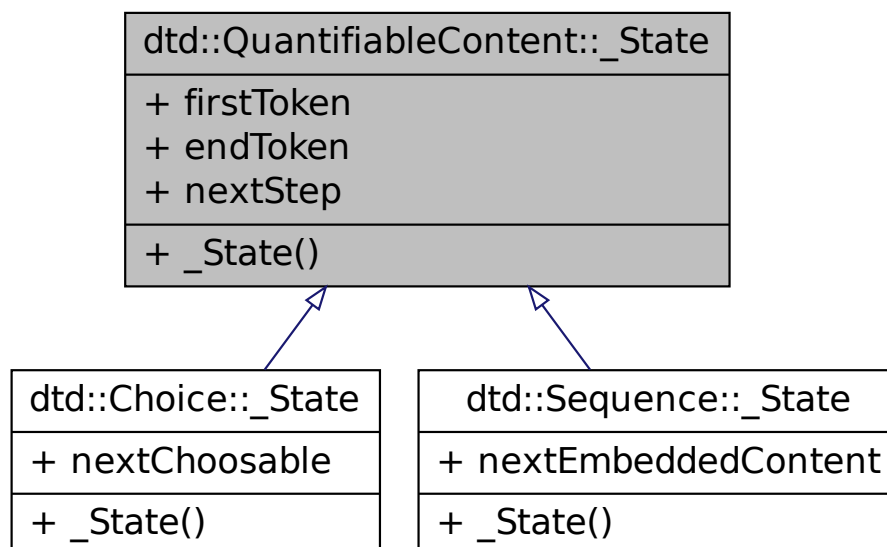
- src/[Choice.hh](#)

## 6.2 Référence de la structure dtd : :QuantifiableContent : :\_State

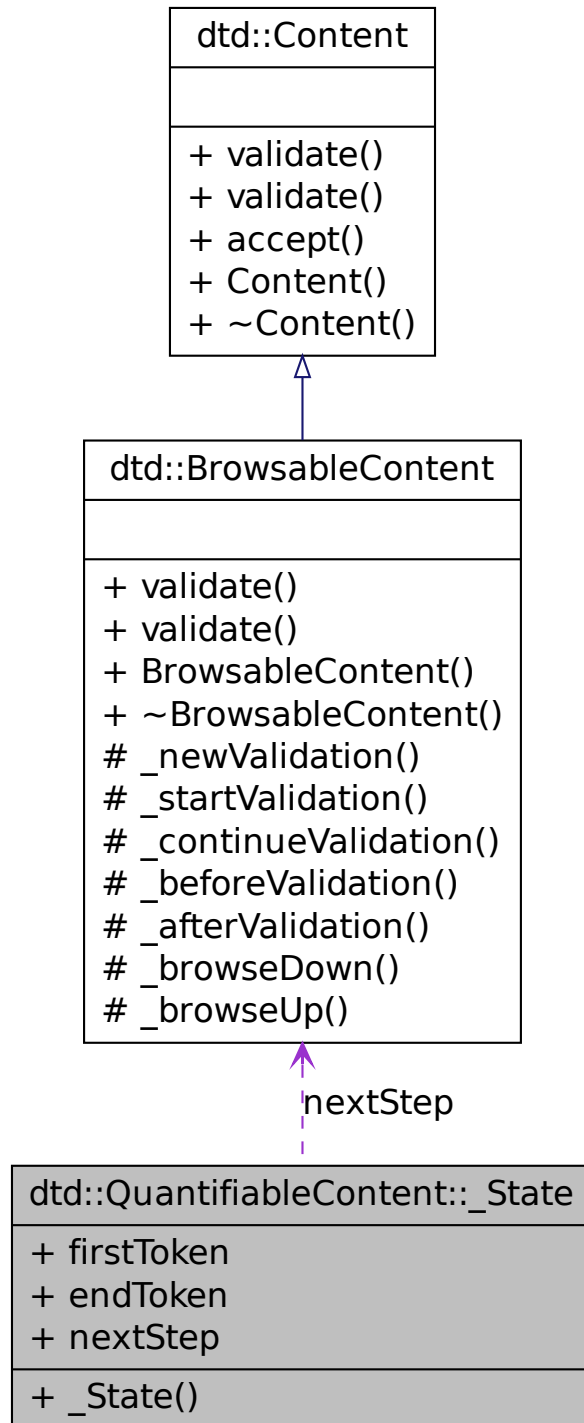
```
#include <QuantifiableContent.hh>
```



Graphe d'héritage de dtd : :QuantifiableContent : :\_State :



Graphe de collaboration de dtd::QuantifiableContent::\_State :



### Fonctions membres publiques

- `_State` (`xml : :CompositeMarkupNode : :ChildrenIterator aFirstToken`, `xml : :CompositeMarkupNode : :ChildrenIterator anEndToken`, `BrowsableContent *aNextStep`)

### Attributs publics

- `xml : :CompositeMarkupNode : :ChildrenIterator firstToken`
- `xml : :CompositeMarkupNode : :ChildrenIterator endToken`
- `BrowsableContent * nextStep`

#### 6.2.1 Description détaillée

Définition à la ligne 50 du fichier `QuantifiableContent.hh`.

#### 6.2.2 Documentation des constructeurs et destructeur

- 6.2.2.1 `dtd : :QuantifiableContent : :State : :State ( xml : :CompositeMarkupNode : :ChildrenIterator aFirstToken, xml : :CompositeMarkupNode : :ChildrenIterator anEndToken, BrowsableContent * aNextStep ) [inline]`

Définition à la ligne 56 du fichier `QuantifiableContent.hh`.

#### 6.2.3 Documentation des données membres

- 6.2.3.1 `xml : :CompositeMarkupNode : :ChildrenIterator dtd : :QuantifiableContent : : _State : :endToken`

Définition à la ligne 53 du fichier `QuantifiableContent.hh`.

- 6.2.3.2 `xml : :CompositeMarkupNode : :ChildrenIterator dtd : :QuantifiableContent : : _State : :firstToken`

Définition à la ligne 52 du fichier `QuantifiableContent.hh`.

- 6.2.3.3 `BrowsableContent* dtd : :QuantifiableContent : : _State : :nextStep`

Définition à la ligne 54 du fichier `QuantifiableContent.hh`.

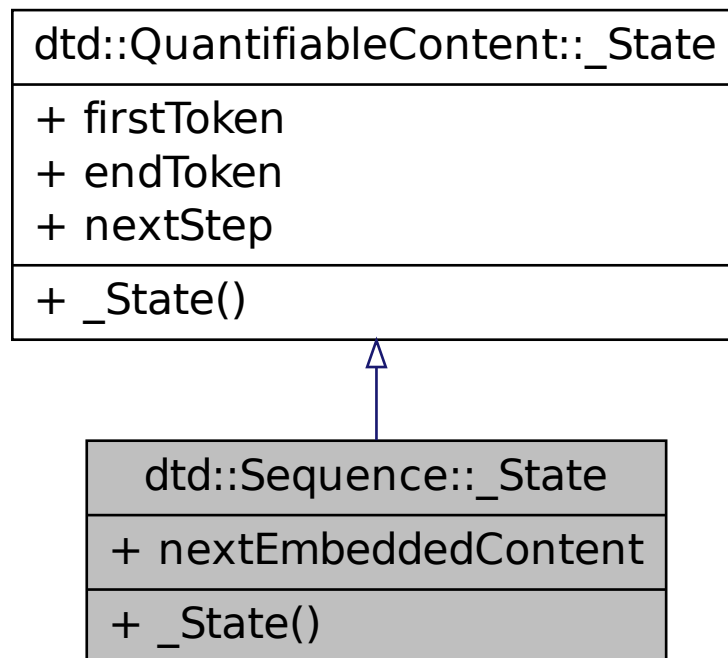
La documentation de cette structure a été générée à partir du fichier suivant :

- `src/QuantifiableContent.hh`

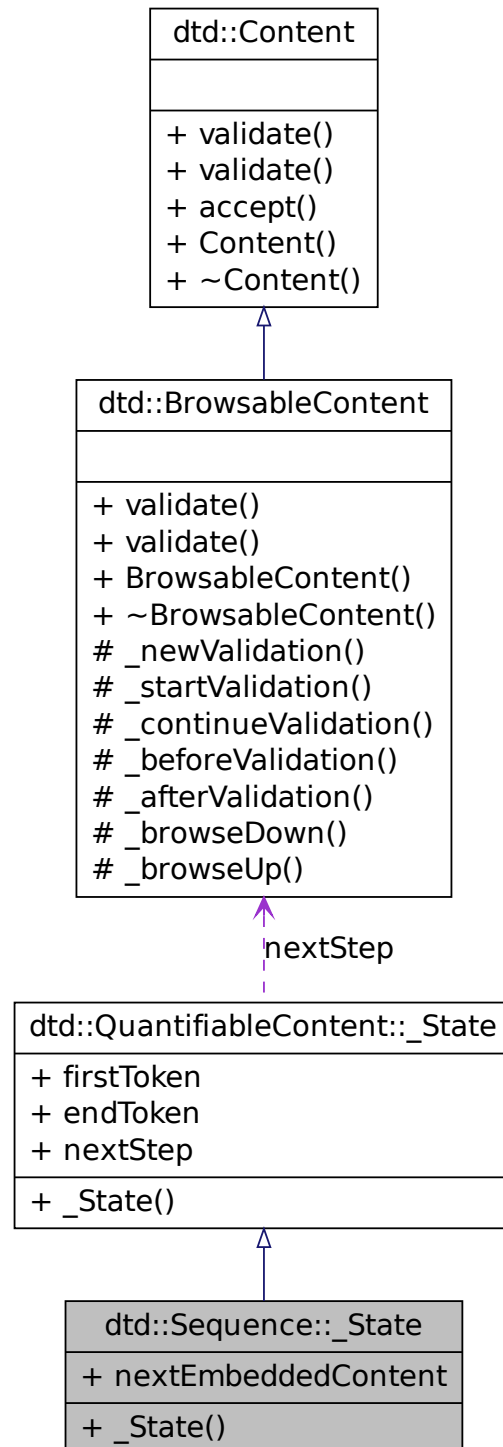
### 6.3 Référence de la structure dtd : :Sequence : : \_State

```
#include <Sequence.hh>
```

Graphe d'héritage de dtd::Sequence::\_State :



Graphe de collaboration de dtd : :Sequence : :\_State :



## Fonctions membres publiques

- `_State` (xml : :CompositeMarkupNode : :ChildrenIterator aFirstToken, xml : :CompositeMarkupNode : :ChildrenIterator anEndToken, [BrowsableContent](#) \*aNextStep, `_OrderedContent` : :iterator aNextEmbeddedContent)

## Attributs publics

- `_OrderedContent` : :iterator [nextEmbeddedContent](#)

### 6.3.1 Description détaillée

Définition à la ligne 83 du fichier Sequence.hh.

### 6.3.2 Documentation des constructeurs et destructeur

6.3.2.1 `dtd : :Sequence : :_State : :_State ( xml : :CompositeMarkupNode : :ChildrenIterator aFirstToken, xml : :CompositeMarkupNode : :ChildrenIterator anEndToken, BrowsableContent * aNextStep, _OrderedContent : :iterator aNextEmbeddedContent )`  
`[inline]`

Définition à la ligne 87 du fichier Sequence.hh.

### 6.3.3 Documentation des données membres

6.3.3.1 `_OrderedContent` : :iterator `dtd : :Sequence : :_State : :nextEmbeddedContent`

Définition à la ligne 85 du fichier Sequence.hh.

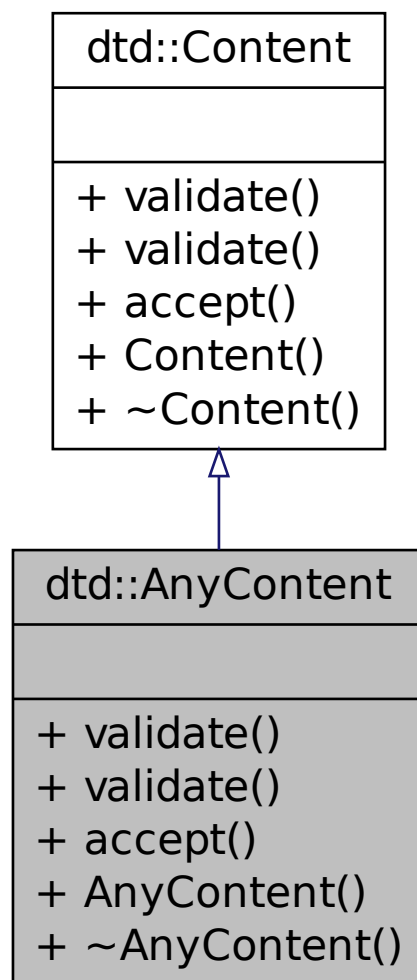
La documentation de cette structure a été générée à partir du fichier suivant :

- src/[Sequence.hh](#)

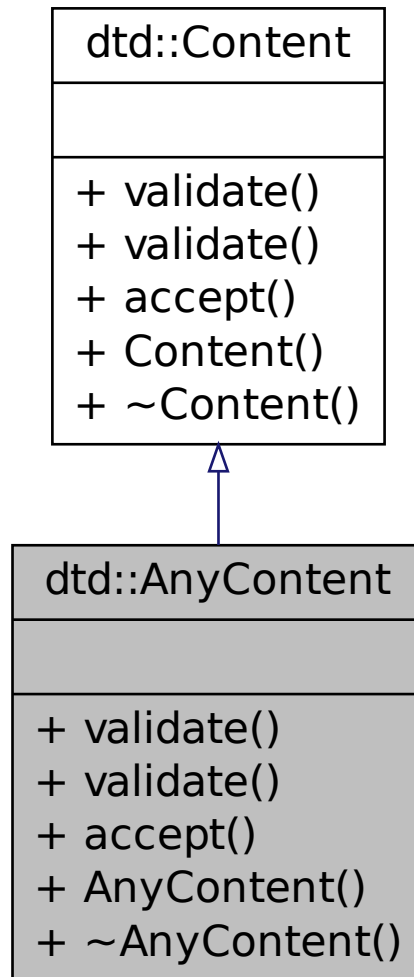
## 6.4 Référence de la classe `dtd : :AnyContent`

```
#include <AnyContent.hh>
```

Graphe d'héritage de dtd : :AnyContent :



Graphe de collaboration de dtd : :AnyContent :



### Fonctions membres publiques

- virtual bool [validate](#) (const xml : :MarkupNode &node)
- virtual bool [validate](#) (const xml : :CompositeMarkupNode &node)
- virtual void [accept](#) ([InterfaceDTDVisitor](#) &visitor) const
- [AnyContent](#) ()
- virtual [~AnyContent](#) ()



### 6.4.1 Description détaillée

Définition à la ligne 18 du fichier AnyContent.hh.

### 6.4.2 Documentation des constructeurs et destructeur

#### 6.4.2.1 dtd : :AnyContent : :AnyContent ( )

Définition à la ligne 52 du fichier AnyContent.cpp.

#### 6.4.2.2 dtd : :AnyContent : :~AnyContent ( ) [virtual]

Définition à la ligne 57 du fichier AnyContent.cpp.

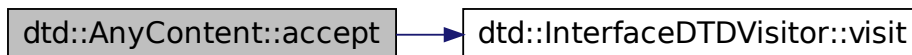
### 6.4.3 Documentation des fonctions membres

#### 6.4.3.1 void dtd : :AnyContent : :accept ( InterfaceDTDVisitor & *visitor* ) const [virtual]

Implémente [dtd : :Content](#).

Définition à la ligne 43 du fichier AnyContent.cpp.

Voici le graphe d'appel pour cette fonction :



#### 6.4.3.2 virtual bool dtd : :AnyContent : :validate ( const xml : :MarkupNode & *node* ) [virtual]

Implémente [dtd : :Content](#).

#### 6.4.3.3 virtual bool dtd : :AnyContent : :validate ( const xml : :CompositeMarkupNode & *node* ) [virtual]

Implémente [dtd : :Content](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

- [src/AnyContent.hh](#)
- [src/AnyContent.cpp](#)

## 6.5 Référence de la classe dtd : :Attribute

```
#include <Attribute.hh>
```

### Fonctions membres publiques

- std : :string [name](#) () const
- [Attribute](#) (const std : :string &name)
- virtual [~Attribute](#) ()

### Attributs protégés

- std : :string [\\_name](#)

#### 6.5.1 Description détaillée

Définition à la ligne 18 du fichier Attribute.hh.

#### 6.5.2 Documentation des constructeurs et destructeur

6.5.2.1 dtd : :Attribute : :Attribute ( const std : :string & *name* )

6.5.2.2 dtd : :Attribute : :~Attribute ( ) [virtual]

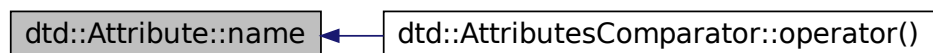
Définition à la ligne 46 du fichier Attribute.cpp.

#### 6.5.3 Documentation des fonctions membres

6.5.3.1 string dtd : :Attribute : :name ( ) const

Définition à la ligne 31 du fichier Attribute.cpp.

Voici le graphe d'appel pour cette fonction :



### 6.5.4 Documentation des données membres

#### 6.5.4.1 std : :string dtd : :Attribute : :\_name [protected]

Définition à la ligne 43 du fichier Attribute.hh.

La documentation de cette classe a été générée à partir des fichiers suivants :

- src/[Attribute.hh](#)
- src/[Attribute.cpp](#)

## 6.6 Référence de la structure dtd : :AttributesComparator

```
#include <AttributesList.hh>
```

### Fonctions membres publiques

- bool [operator\(\)](#) (const [Attribute](#) \*const &x, const [Attribute](#) \*const &y) const

#### 6.6.1 Description détaillée

Définition à la ligne 20 du fichier AttributesList.hh.

#### 6.6.2 Documentation des fonctions membres

##### 6.6.2.1 bool dtd : :AttributesComparator : :operator() ( const Attribute \*const & x, const Attribute \*const & y ) const [inline]

Définition à la ligne 22 du fichier AttributesList.hh.

Voici le graphe d'appel pour cette fonction :



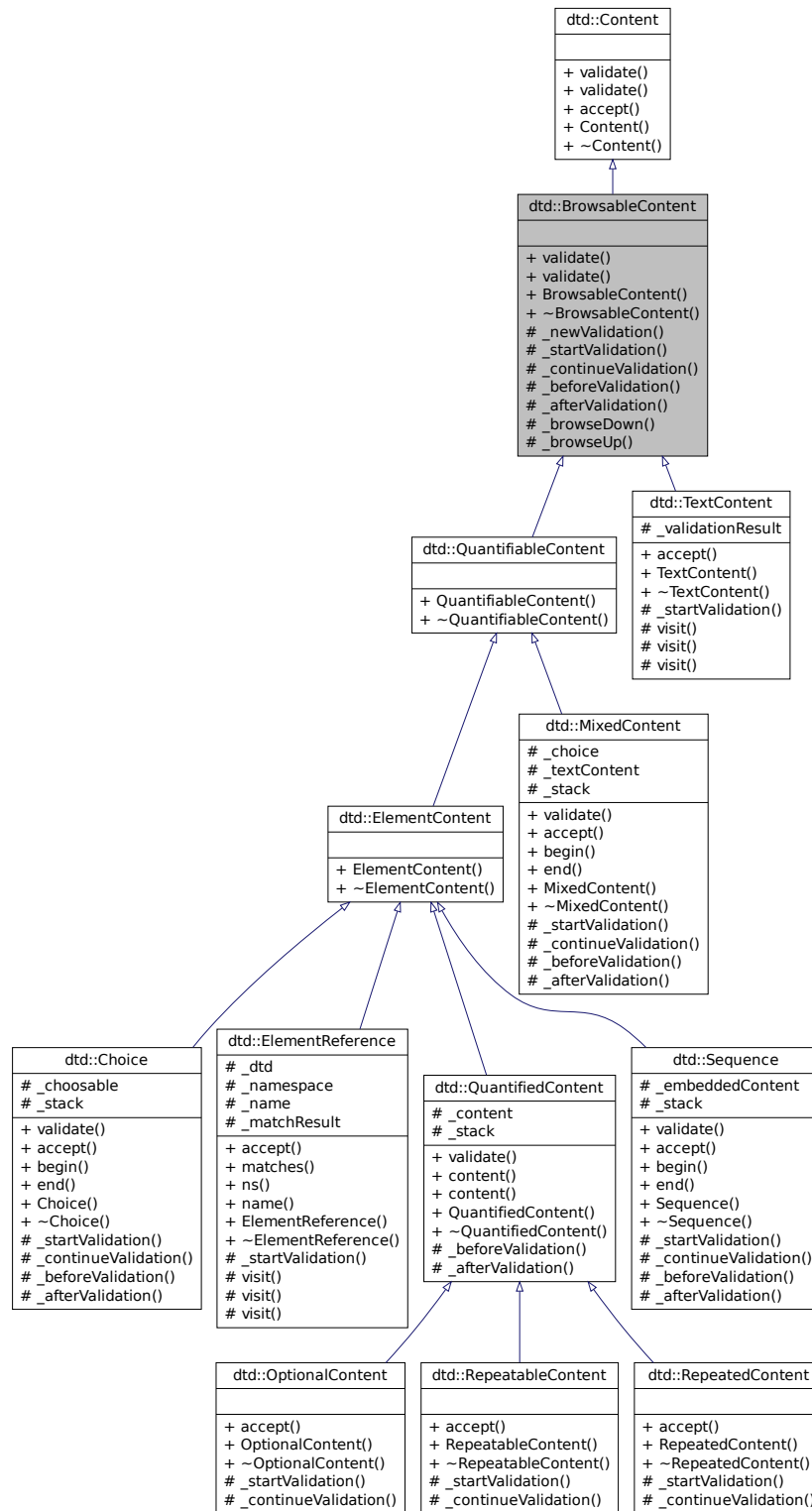
La documentation de cette structure a été générée à partir du fichier suivant :

- src/[AttributesList.hh](#)

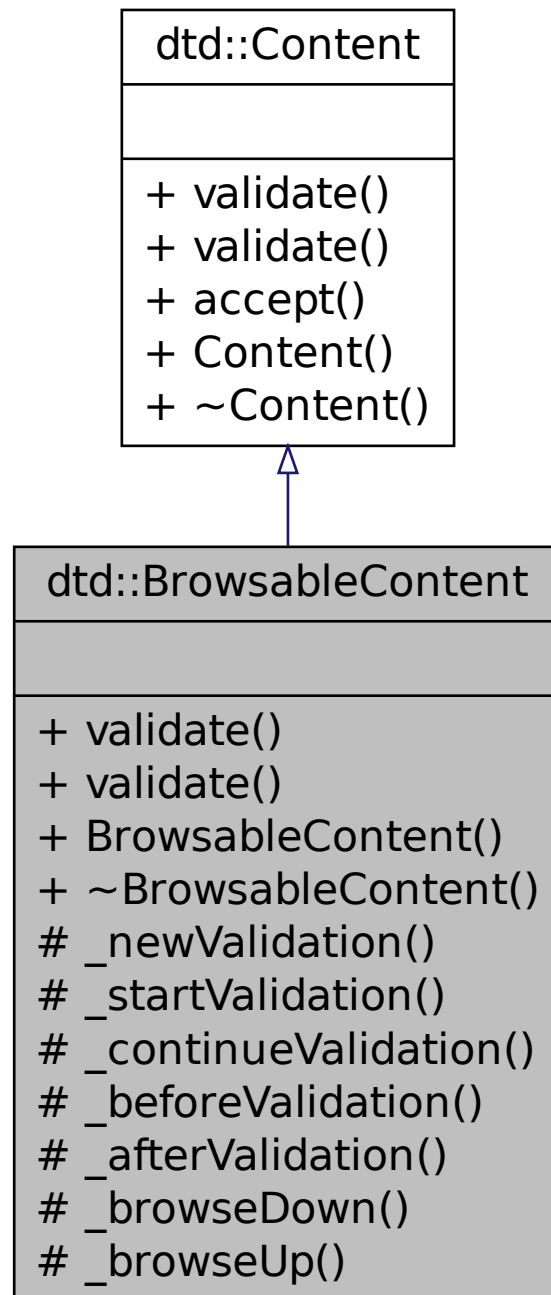
## 6.7 Référence de la classe dtd : :BrowsableContent

```
#include <BrowsableContent.hh>
```

Graphe d'héritage de dtd : :BrowsableContent :



Graphe de collaboration de dtd : :BrowsableContent :



## Fonctions membres publiques

- virtual bool [validate](#) (const xml : :MarkupNode &node)
- virtual bool [validate](#) (const xml : :CompositeMarkupNode &node)
- [BrowsableContent](#) ()
- virtual ~[BrowsableContent](#) ()

## Fonctions membres protégées

- bool [\\_newValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- virtual bool [\\_startValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)=0
- virtual bool [\\_continueValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator currentToken)
- virtual void [\\_beforeValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- virtual void [\\_afterValidation](#) ()
- bool [\\_browseDown](#) ([BrowsableContent](#) &childContent, xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- bool [\\_browseUp](#) ([BrowsableContent](#) \*parentContent, xml : :CompositeMarkupNode : :ChildrenIterator currentToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken)

### 6.7.1 Description détaillée

Définition à la ligne 18 du fichier BrowsableContent.hh.

### 6.7.2 Documentation des constructeurs et destructeur

#### 6.7.2.1 dtd : :BrowsableContent : :BrowsableContent ( )

Définition à la ligne 52 du fichier BrowsableContent.cpp.

#### 6.7.2.2 dtd : :BrowsableContent : :~BrowsableContent ( ) [virtual]

Définition à la ligne 57 du fichier BrowsableContent.cpp.

### 6.7.3 Documentation des fonctions membres

#### 6.7.3.1 void dtd : :BrowsableContent : :afterValidation ( ) [protected, virtual]

Réimplémentée dans [dtd : :Choice](#), [dtd : :MixedContent](#), [dtd : :QuantifiedContent](#), et [dtd : :Sequence](#).

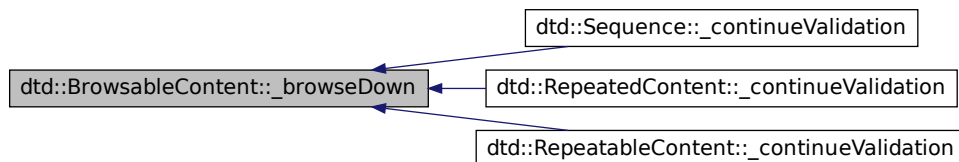
Définition à la ligne 107 du fichier BrowsableContent.cpp.

6.7.3.2 virtual void dtd : :BrowsableContent : :\_beforeValidation (   
 xml : :CompositeMarkupNode : :ChildrenIterator *firstToken*,   
 xml : :CompositeMarkupNode : :ChildrenIterator *endToken*, BrowsableContent \*   
*nextStep* ) [protected, virtual]

Réimplémentée dans [dtd : :Choice](#), [dtd : :MixedContent](#), [dtd : :QuantifiedContent](#), et [dtd : :Sequence](#).

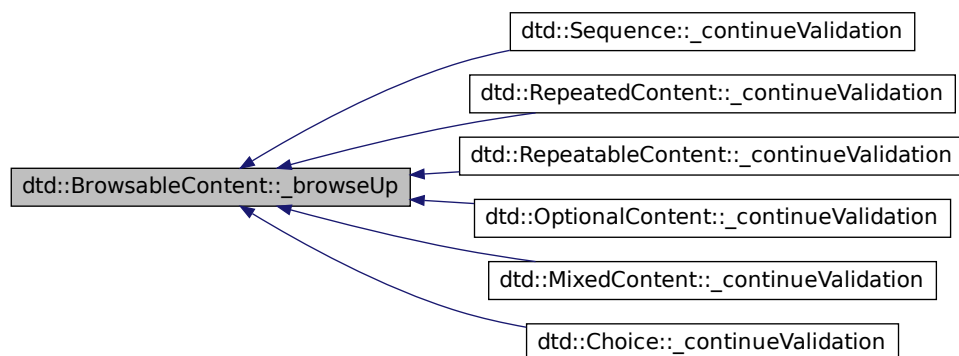
6.7.3.3 bool dtd : :BrowsableContent : :\_browseDown ( BrowsableContent &   
*childContent*, xml : :CompositeMarkupNode : :ChildrenIterator *firstToken*,   
 xml : :CompositeMarkupNode : :ChildrenIterator *endToken*, BrowsableContent \*   
*nextStep* ) [protected]

Voici le graphe d'appel pour cette fonction :



6.7.3.4 bool dtd : :BrowsableContent : :\_browseUp ( BrowsableContent \*   
*parentContent*, xml : :CompositeMarkupNode : :ChildrenIterator *currentToken*,   
 xml : :CompositeMarkupNode : :ChildrenIterator *endToken* ) [protected]

Voici le graphe d'appel pour cette fonction :



**6.7.3.5** `bool dtd : :BrowsableContent : :_continueValidation (`  
`xml : :CompositeMarkupNode : :ChildrenIterator currentToken )` `[protected,`  
`virtual]`

Réimplémentée dans [dtd : :Choice](#), [dtd : :MixedContent](#), [dtd : :OptionalContent](#), [dtd : :RepeatableContent](#), [dtd : :RepeatedContent](#), et [dtd : :Sequence](#).

Définition à la ligne 92 du fichier `BrowsableContent.cpp`.

**6.7.3.6** `bool dtd : :BrowsableContent : :_newValidation (`  
`xml : :CompositeMarkupNode : :ChildrenIterator firstToken,`  
`xml : :CompositeMarkupNode : :ChildrenIterator endToken, BrowsableContent *`  
`nextStep )` `[protected]`

**6.7.3.7** `virtual bool dtd : :BrowsableContent : :_startValidation (`  
`xml : :CompositeMarkupNode : :ChildrenIterator firstToken,`  
`xml : :CompositeMarkupNode : :ChildrenIterator endToken, BrowsableContent *`  
`nextStep )` `[protected, pure virtual]`

Implémenté dans [dtd : :Choice](#), [dtd : :ElementReference](#), [dtd : :MixedContent](#), [dtd : :OptionalContent](#), [dtd : :RepeatableContent](#), [dtd : :RepeatedContent](#), [dtd : :Sequence](#), et [dtd : :TextContent](#).

**6.7.3.8** `virtual bool dtd : :BrowsableContent : :validate ( const xml : :MarkupNode & node )`  
`[virtual]`

Implémente [dtd : :Content](#).

**6.7.3.9** `virtual bool dtd : :BrowsableContent : :validate ( const xml : :CompositeMarkupNode &`  
`node )` `[virtual]`

Implémente [dtd : :Content](#).

Réimplémentée dans [dtd : :Choice](#), [dtd : :MixedContent](#), [dtd : :QuantifiedContent](#), et [dtd : :Sequence](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

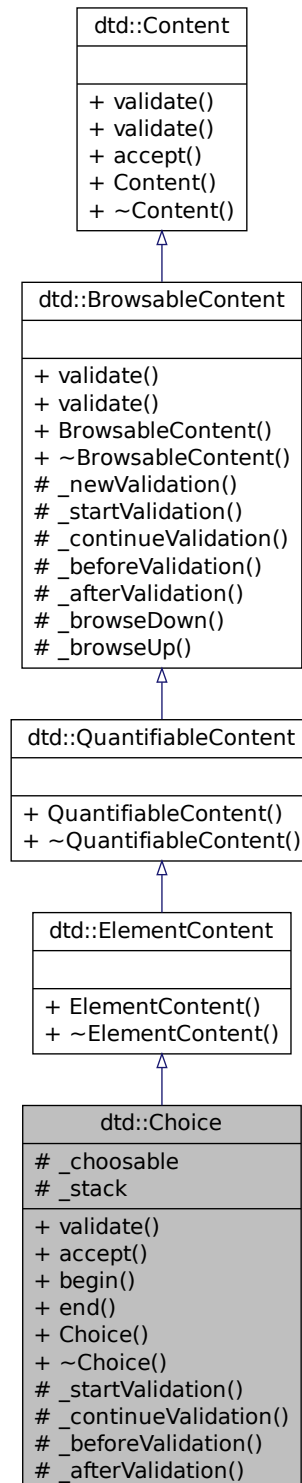
- [src/BrowsableContent.hh](#)
- [src/BrowsableContent.cpp](#)

## 6.8 Référence de la classe `dtd : :Choice`

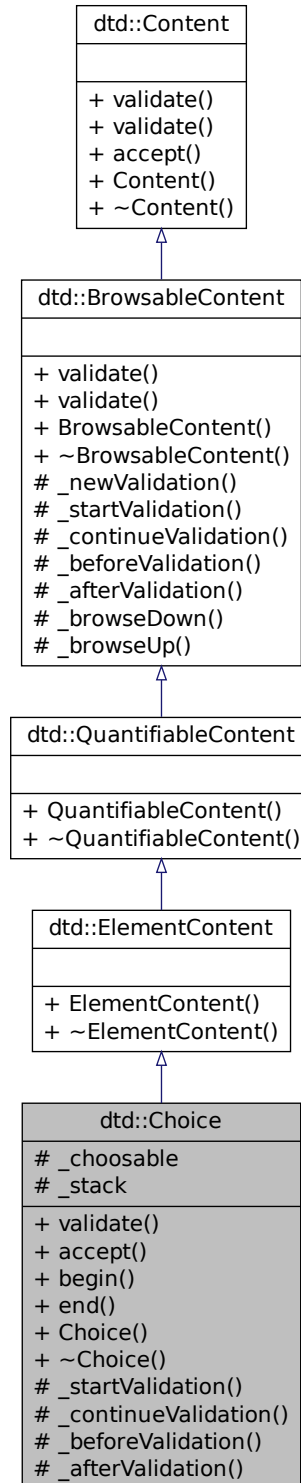
```
#include <Choice.hh>
```



Graphe d'héritage de dtd : :Choice :



Graphe de collaboration de dtd : :Choice :



## Classes

- struct [\\_State](#)

## Types publics

- typedef std : :set< [ElementContent](#) \* > [ChoosableSet](#)
- typedef [\\_ChoosableSet](#) : :const\_iterator [const\\_iterator](#)

## Fonctions membres publiques

- virtual bool [validate](#) (const xml : :CompositeMarkupNode &node)
- virtual void [accept](#) ([InterfaceDTDVisitor](#) &visitor) const
- [const\\_iterator](#) [begin](#) () const
- [const\\_iterator](#) [end](#) () const
- [Choice](#) (const [ChoosableSet](#) &elements)
- virtual [~Choice](#) ()

## Types protégés

- typedef std : :set< [ElementContent](#) \* > [\\_ChoosableSet](#)
- typedef std : :stack< [\\_State](#) > [\\_StatesStack](#)

## Fonctions membres protégées

- virtual bool [\\_startValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- virtual bool [\\_continueValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator currentToken)
- virtual void [\\_beforeValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- virtual void [\\_afterValidation](#) ()

## Attributs protégés

- [\\_ChoosableSet](#) [\\_choosable](#)
- [\\_StatesStack](#) [\\_stack](#)

### 6.8.1 Description détaillée

Définition à la ligne 20 du fichier Choice.hh.

## 6.8.2 Documentation des définitions de type membres

**6.8.2.1** `typedef std::set<ElementContent*> dtd::Choice::_ChoosableSet`  
`[protected]`

Définition à la ligne 23 du fichier Choice.hh.

**6.8.2.2** `typedef std::stack<_State> dtd::Choice::_StatesStack` `[protected]`

Définition à la ligne 92 du fichier Choice.hh.

**6.8.2.3** `typedef std::set<ElementContent*> dtd::Choice::ChoosableSet`

Définition à la ligne 28 du fichier Choice.hh.

**6.8.2.4** `typedef _ChoosableSet::const_iterator dtd::Choice::const_iterator`

Définition à la ligne 29 du fichier Choice.hh.

## 6.8.3 Documentation des constructeurs et destructeur

**6.8.3.1** `dtd::Choice::Choice ( const ChoosableSet & elements )`

Définition à la ligne 59 du fichier Choice.cpp.

**6.8.3.2** `dtd::Choice::~~Choice ( )` `[virtual]`

Définition à la ligne 65 du fichier Choice.cpp.

## 6.8.4 Documentation des fonctions membres

**6.8.4.1** `void dtd::Choice::_afterValidation ( )` `[protected, virtual]`

Réimplémentée à partir de [dtd::BrowsableContent](#).

Définition à la ligne 85 du fichier Choice.cpp.

**6.8.4.2** `void dtd::Choice::_beforeValidation ( xml::CompositeMarkupNode::ChildrenIterator`  
`firstToken, xml::CompositeMarkupNode::ChildrenIterator endToken,`  
`BrowsableContent * nextStep )` `[protected, virtual]`

Réimplémentée à partir de [dtd::BrowsableContent](#).

Définition à la ligne 77 du fichier Choice.cpp.

**6.8.4.3** `bool dtd : :Choice : :continueValidation ( xml : :CompositeMarkupNode : :ChildrenIterator  
currentToken ) [protected, virtual]`

Réimplémentée à partir de [dtd : :BrowsableContent](#).

Définition à la ligne 109 du fichier Choice.cpp.

Voici le graphe d'appel pour cette fonction :



**6.8.4.4** `virtual bool dtd : :Choice : :startValidation ( xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken,  
BrowsableContent * nextStep ) [protected, virtual]`

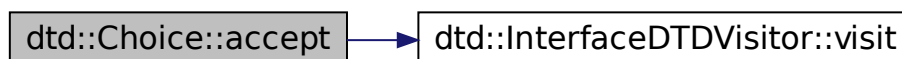
Implémente [dtd : :BrowsableContent](#).

**6.8.4.5** `void dtd : :Choice : :accept ( InterfaceDTDVisitor & visitor ) const [virtual]`

Implémente [dtd : :Content](#).

Définition à la ligne 40 du fichier Choice.cpp.

Voici le graphe d'appel pour cette fonction :



**6.8.4.6** `Choice : :const_iterator dtd : :Choice : :begin ( ) const`

Définition à la ligne 45 du fichier Choice.cpp.

**6.8.4.7** `Choice : :const_iterator dtd : :Choice : :end ( ) const`

Définition à la ligne 50 du fichier Choice.cpp.

**6.8.4.8** `virtual bool dtd : :Choice : :validate ( const xml : :CompositeMarkupNode & node )`  
`[virtual]`

Réimplémentée à partir de [dtd : :BrowsableContent](#).

## 6.8.5 Documentation des données membres

**6.8.5.1** `_ChoosableSet dtd : :Choice : :_choosable` `[protected]`

Définition à la ligne 76 du fichier `Choice.hh`.

**6.8.5.2** `_StatesStack dtd : :Choice : :_stack` `[protected]`

Définition à la ligne 93 du fichier `Choice.hh`.

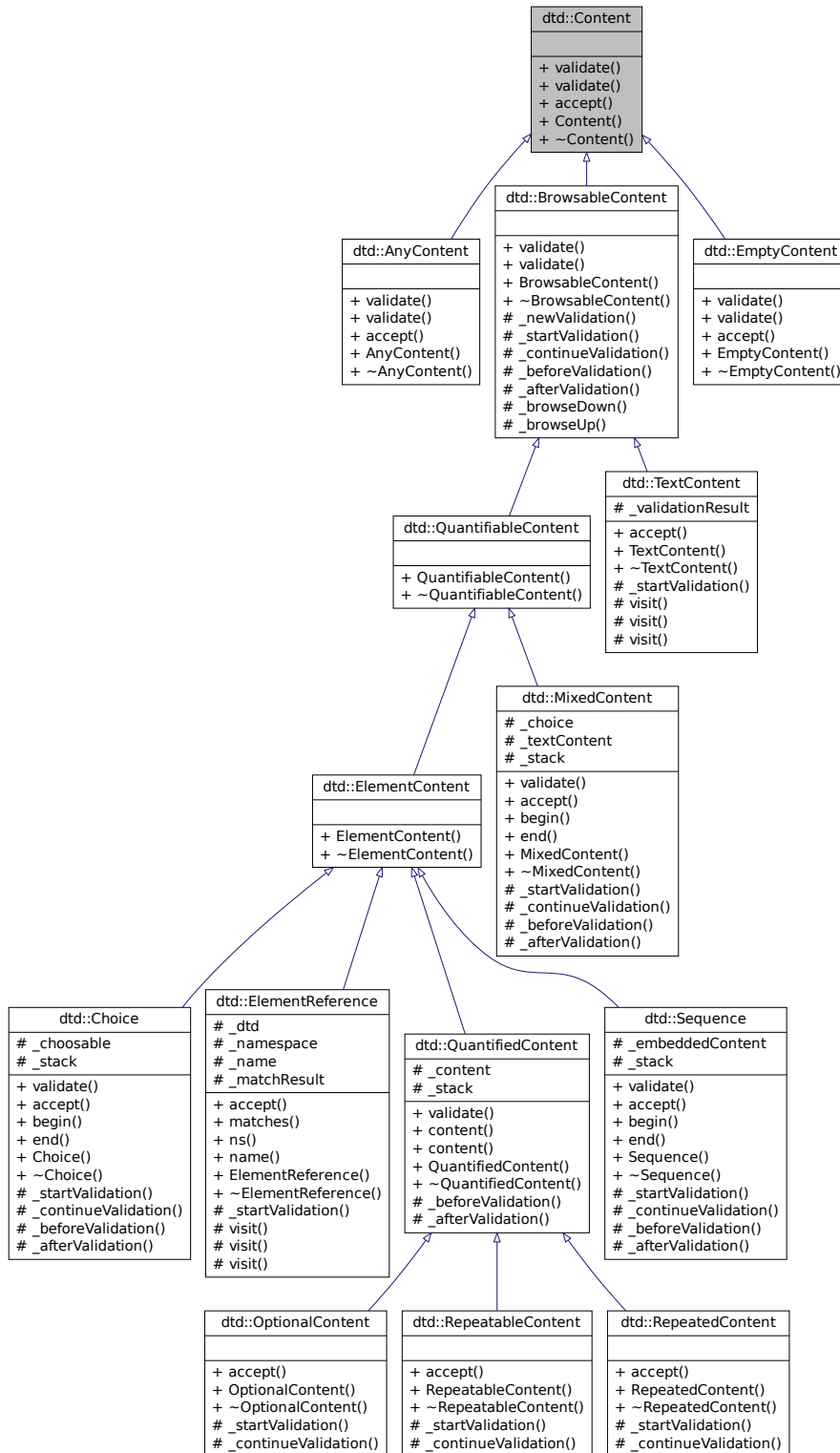
La documentation de cette classe a été générée à partir des fichiers suivants :

- [src/Choice.hh](#)
- [src/Choice.cpp](#)

## 6.9 Référence de la classe `dtd : :Content`

```
#include <Content.hh>
```

Graphe d'héritage de dtd : :Content :



## Fonctions membres publiques

- virtual bool [validate](#) (const xml : :MarkupNode &node)=0
- virtual bool [validate](#) (const xml : :CompositeMarkupNode &node)=0
- virtual void [accept](#) ([InterfaceDTDVisitor](#) &visitor) const =0
- [Content](#) ()
- virtual [~Content](#) ()

### 6.9.1 Description détaillée

Définition à la ligne 21 du fichier Content.hh.

### 6.9.2 Documentation des constructeurs et destructeur

#### 6.9.2.1 `dtd : :Content : :Content ( )`

Définition à la ligne 38 du fichier Content.cpp.

#### 6.9.2.2 `dtd : :Content : :~Content ( )` [virtual]

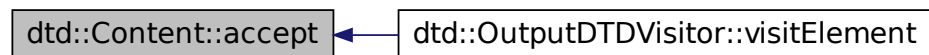
Définition à la ligne 43 du fichier Content.cpp.

### 6.9.3 Documentation des fonctions membres

#### 6.9.3.1 `virtual void dtd : :Content : :accept ( InterfaceDTDVisitor & visitor ) const` [pure virtual]

Implémenté dans [dtd : :AnyContent](#), [dtd : :Choice](#), [dtd : :ElementReference](#), [dtd : :EmptyContent](#), [dtd : :MixedContent](#), [dtd : :OptionalContent](#), [dtd : :RepeatableContent](#), [dtd : :RepeatedContent](#), [dtd : :Sequence](#), et [dtd : :TextContent](#).

Voici le graphe d'appel pour cette fonction :



#### 6.9.3.2 `virtual bool dtd : :Content : :validate ( const xml : :MarkupNode & node )` [pure virtual]

Implémenté dans [dtd : :AnyContent](#), [dtd : :BrowsableContent](#), et [dtd : :EmptyContent](#).



**6.9.3.3** virtual bool dtd : :Content : :validate ( const xml : :CompositeMarkupNode & node )  
[pure virtual]

Implémenté dans [dtd : :AnyContent](#), [dtd : :BrowsableContent](#), [dtd : :Choice](#), [dtd : :EmptyContent](#), [dtd : :MixedContent](#), [dtd : :QuantifiedContent](#), et [dtd : :Sequence](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

- src/[Content.hh](#)
- src/[Content.cpp](#)

## 6.10 Référence de la classe dtd : :DTD

```
#include <DTD.hh>
```

### Fonctions membres publiques

- void [accept](#) ([InterfaceDTDVisitor](#) &visitor)
- void [addElement](#) (const std : :string &ns, const std : :string &elementName, [Content](#) &content)
- void [addAttributesList](#) (const std : :string &ns, const std : :string &elementName, const [AttributesList](#) &attribute)
- bool [isValid](#) (const xml : :Node &root, const std : :string &validRootName)
- [DTD](#) ()
- virtual [~DTD](#) ()

### Types protégés

- typedef std : :pair< std : :string, std : :string > [\\_ElementId](#)
- typedef std : :map< [\\_ElementId](#), [Content](#) \* > [\\_Elements](#)
- typedef std : :map< [\\_ElementId](#), [AttributesList](#) > [\\_AttributesLists](#)

### Fonctions membres protégées

- virtual void [visit](#) (const xml : :TextNode &node)
- virtual void [visit](#) (const xml : :MarkupNode &node)
- virtual void [visit](#) (const xml : :CompositeMarkupNode &node)
- bool [checkAttributes](#) (const xml : :MarkupNode &node)
- [Content](#) \* [getElement](#) (std : :string ns, std : :string name) const
- const [AttributesList](#) \* [getAttributesList](#) (std : :string ns, std : :string name) const
- bool [\\_isValid](#) (const xml : :Node &node)

### Attributs protégés

- [\\_Elements](#) [elements](#)
- [\\_AttributesLists](#) [attributesLists](#)
- bool [\\_validatingRoot](#)
- std : :string [\\_validRootName](#)
- bool [\\_lastNodeIsValid](#)

### 6.10.1 Description détaillée

Définition à la ligne 24 du fichier DTD.hh.

### 6.10.2 Documentation des définitions de type membres

**6.10.2.1** `typedef std : :map<_ElementId, AttributesList> dtd : :DTD : :_AttributesLists  
[protected]`

Définition à la ligne 81 du fichier DTD.hh.

**6.10.2.2** `typedef std : :pair<std : :string, std : :string> dtd : :DTD : :_ElementId  
[protected]`

Définition à la ligne 79 du fichier DTD.hh.

**6.10.2.3** `typedef std : :map<_ElementId, Content*> dtd : :DTD : :_Elements  
[protected]`

Définition à la ligne 80 du fichier DTD.hh.

### 6.10.3 Documentation des constructeurs et destructeur

**6.10.3.1** `dtd : :DTD : :DTD ( )`

Définition à la ligne 134 du fichier DTD.cpp.

**6.10.3.2** `dtd : :DTD : :~DTD ( ) [virtual]`

Définition à la ligne 141 du fichier DTD.cpp.

### 6.10.4 Documentation des fonctions membres

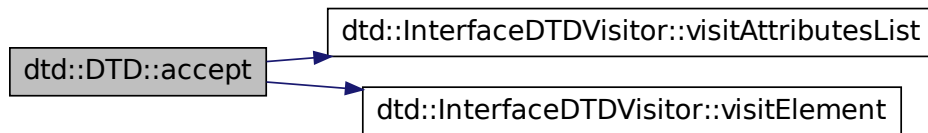
**6.10.4.1** `bool dtd : :DTD : :IsValid ( const xml : :Node & node ) [protected]`

Définition à la ligne 161 du fichier DTD.cpp.

**6.10.4.2** `void dtd : :DTD : :accept ( InterfaceDTDVisitor & visitor )`

Définition à la ligne 41 du fichier DTD.cpp.

Voici le graphe d'appel pour cette fonction :



**6.10.4.3** `void dtd : :DTD : :addAttributesList ( const std : :string & ns, const std : :string & elementName, const AttributesList & attribute )`

Définition à la ligne 71 du fichier DTD.cpp.

**6.10.4.4** `void dtd : :DTD : :addElement ( const std : :string & ns, const std : :string & elementName, Content & content )`

Définition à la ligne 57 du fichier DTD.cpp.

**6.10.4.5** `bool dtd : :DTD : :checkAttributes ( const xml : :MarkupNode & node )`  
[protected]

Définition à la ligne 250 du fichier DTD.cpp.

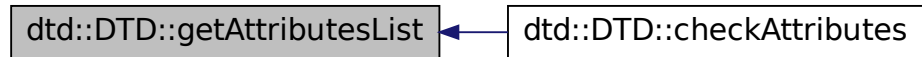
Voici le graphe d'appel pour cette fonction :



**6.10.4.6** `const AttributesList * dtd : :DTD : :getAttributesList ( std : :string ns, std : :string name ) const` [protected]

Définition à la ligne 107 du fichier DTD.cpp.

Voici le graphe d'appel pour cette fonction :



**6.10.4.7** Content \* `dtd : : DTD : : getElement ( std : : string ns, std : : string name ) const`  
[protected]

Définition à la ligne 92 du fichier DTD.cpp.

**6.10.4.8** `bool dtd : : DTD : : isValid ( const xml : : Node & root, const std : : string & validRootName )`

Définition à la ligne 123 du fichier DTD.cpp.

**6.10.4.9** `virtual void dtd : : DTD : : visit ( const xml : : CompositeMarkupNode & node )`  
[protected, virtual]

**6.10.4.10** `virtual void dtd : : DTD : : visit ( const xml : : TextNode & node )` [protected, virtual]

**6.10.4.11** `virtual void dtd : : DTD : : visit ( const xml : : MarkupNode & node )` [protected, virtual]

## 6.10.5 Documentation des données membres

**6.10.5.1** `_AttributesLists dtd : : DTD : : _attributesLists` [protected]

Définition à la ligne 84 du fichier DTD.hh.

**6.10.5.2** `_Elements dtd : : DTD : : _elements` [protected]

Définition à la ligne 83 du fichier DTD.hh.

**6.10.5.3** `bool dtd : : DTD : : _lastNodesValid` [protected]

Définition à la ligne 88 du fichier DTD.hh.

#### 6.10.5.4 `bool dtd : :DTD : :_validatingRoot` [protected]

Définition à la ligne 86 du fichier DTD.hh.

#### 6.10.5.5 `std : :string dtd : :DTD : :_validRootName` [protected]

Définition à la ligne 87 du fichier DTD.hh.

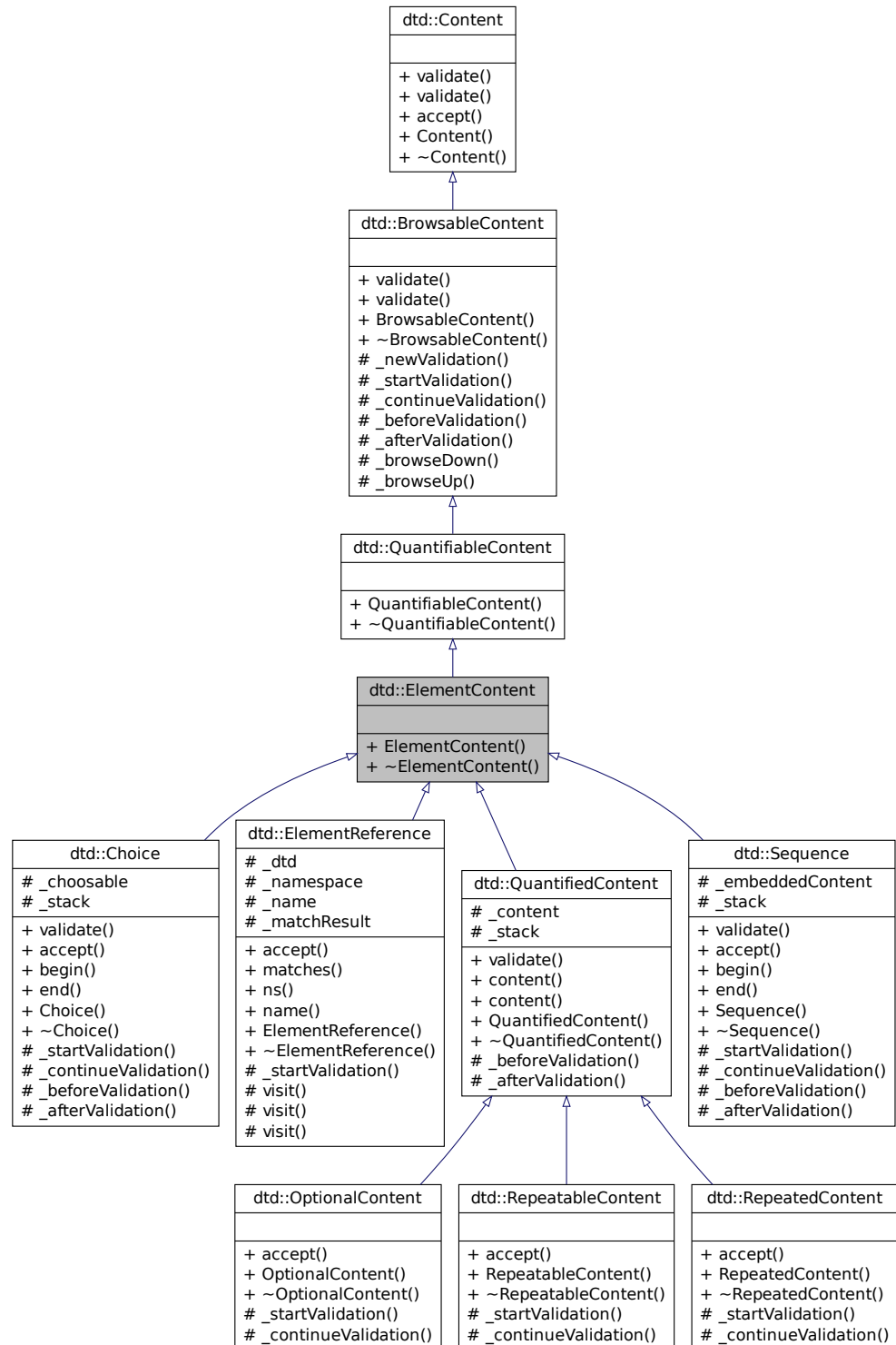
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/[DTD.hh](#)
- src/[DTD.cpp](#)

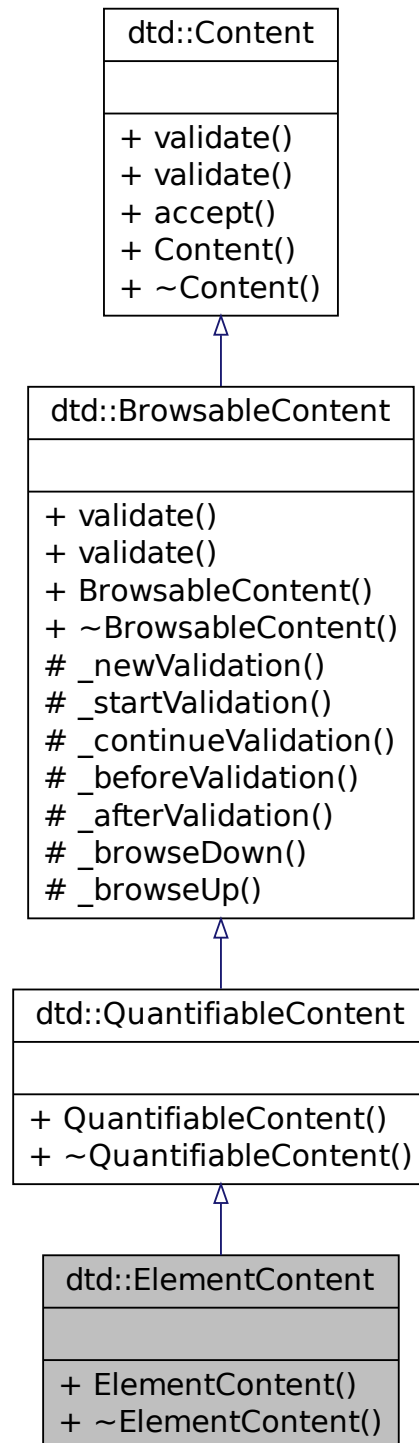
## 6.11 Référence de la classe dtd : :ElementContent

```
#include <ElementContent.hh>
```

Graphe d'héritage de dtd : :ElementContent :



Graphe de collaboration de dtd : :ElementContent :



## Fonctions membres publiques

- [ElementContent](#) ()
- virtual [~ElementContent](#) ()

### 6.11.1 Description détaillée

Définition à la ligne 18 du fichier ElementContent.hh.

### 6.11.2 Documentation des constructeurs et destructeur

#### 6.11.2.1 dtd : :ElementContent : :ElementContent ( )

Définition à la ligne 36 du fichier ElementContent.cpp.

#### 6.11.2.2 dtd : :ElementContent : :~ElementContent ( ) [virtual]

Définition à la ligne 42 du fichier ElementContent.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :

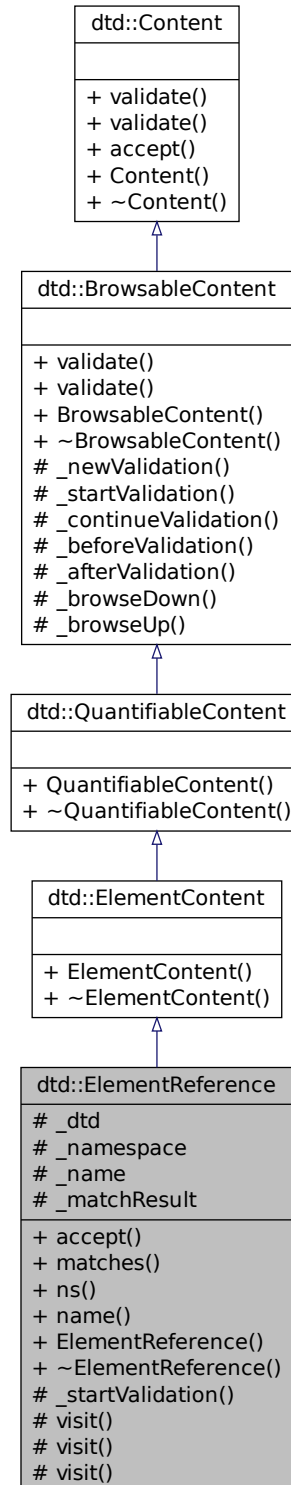
- src/[ElementContent.hh](#)
- src/[ElementContent.cpp](#)

## 6.12 Référence de la classe dtd : :ElementReference

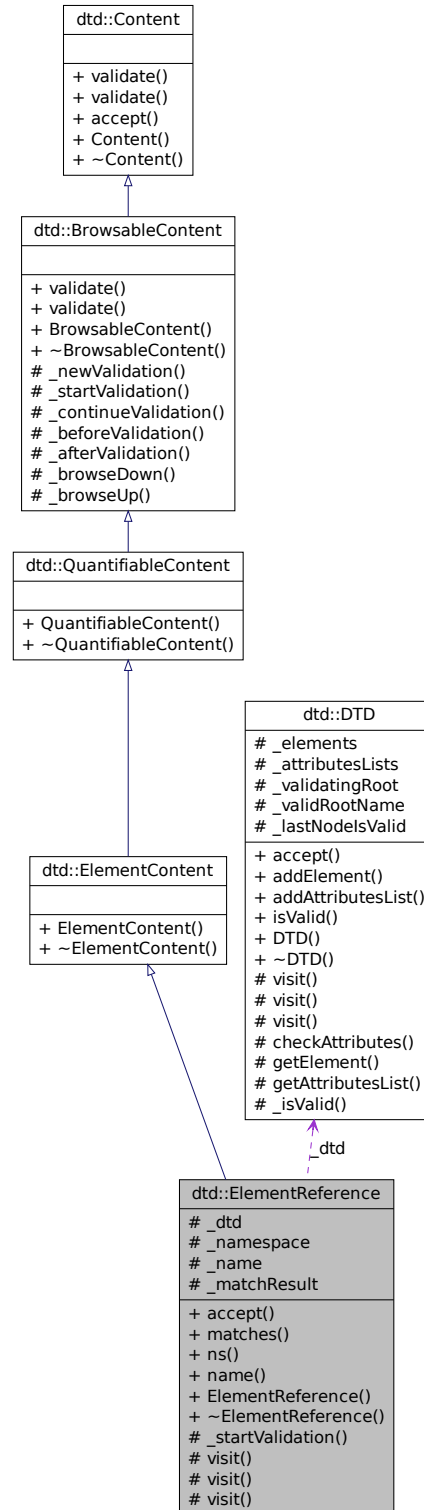
```
#include <ElementReference.hh>
```



Graphe d'héritage de dtd : :ElementReference :



Graphes de collaboration de dtd : :ElementReference :



### Fonctions membres publiques

- virtual void [accept](#) ([InterfaceDTDVisitor](#) &visitor) const
- virtual bool [matches](#) (xml : :Node &node)
- virtual std : :string [ns](#) () const
- virtual std : :string [name](#) () const
- [ElementReference](#) (DTD &dtd, std : :string ns, std : :string name)
- virtual [~ElementReference](#) ()

### Fonctions membres protégées

- virtual bool [\\_startValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- virtual void [visit](#) (const xml : :TextNode &node)
- virtual void [visit](#) (const xml : :MarkupNode &node)
- virtual void [visit](#) (const xml : :CompositeMarkupNode &node)

### Attributs protégés

- [DTD](#) & [\\_dtd](#)
- std : :string [\\_namespace](#)
- std : :string [\\_name](#)
- bool [\\_matchResult](#)

#### 6.12.1 Description détaillée

Définition à la ligne 22 du fichier ElementReference.hh.

#### 6.12.2 Documentation des constructeurs et destructeur

**6.12.2.1** `dtd : :ElementReference : :ElementReference ( DTD & dtd, std : :string ns, std : :string name )`

Définition à la ligne 62 du fichier ElementReference.cpp.

**6.12.2.2** `dtd : :ElementReference : :~ElementReference ( ) [virtual]`

Définition à la ligne 68 du fichier ElementReference.cpp.

#### 6.12.3 Documentation des fonctions membres

**6.12.3.1** `virtual bool dtd : :ElementReference : :_startValidation ( xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, BrowsableContent *nextStep ) [protected, virtual]`

Implémente [dtd : :BrowsableContent](#).

**6.12.3.2** `void dtd : :ElementReference : :accept ( InterfaceDTDVisitor & visitor ) const`  
`[virtual]`

Implémente [dtd : :Content](#).

Définition à la ligne 36 du fichier ElementReference.cpp.

Voici le graphe d'appel pour cette fonction :



**6.12.3.3** `bool dtd : :ElementReference : :matches ( xml : :Node & node )` `[virtual]`

Définition à la ligne 41 du fichier ElementReference.cpp.

**6.12.3.4** `std : :string dtd : :ElementReference : :name ( ) const` `[virtual]`

Définition à la ligne 53 du fichier ElementReference.cpp.

**6.12.3.5** `std : :string dtd : :ElementReference : :ns ( ) const` `[virtual]`

Définition à la ligne 48 du fichier ElementReference.cpp.

**6.12.3.6** `virtual void dtd : :ElementReference : :visit ( const xml : :MarkupNode & node )`  
`[protected, virtual]`

**6.12.3.7** `virtual void dtd : :ElementReference : :visit ( const xml : :CompositeMarkupNode & node )`  
`[protected, virtual]`

**6.12.3.8** `virtual void dtd : :ElementReference : :visit ( const xml : :TextNode & node )`  
`[protected, virtual]`

## 6.12.4 Documentation des données membres

**6.12.4.1** `DTD& dtd : :ElementReference : :_dtd` `[protected]`

Définition à la ligne 54 du fichier ElementReference.hh.

#### 6.12.4.2 `bool dtd : :ElementReference : :_matchResult` [protected]

Définition à la ligne 58 du fichier ElementReference.hh.

#### 6.12.4.3 `std : :string dtd : :ElementReference : :_name` [protected]

Définition à la ligne 56 du fichier ElementReference.hh.

#### 6.12.4.4 `std : :string dtd : :ElementReference : :_namespace` [protected]

Définition à la ligne 55 du fichier ElementReference.hh.

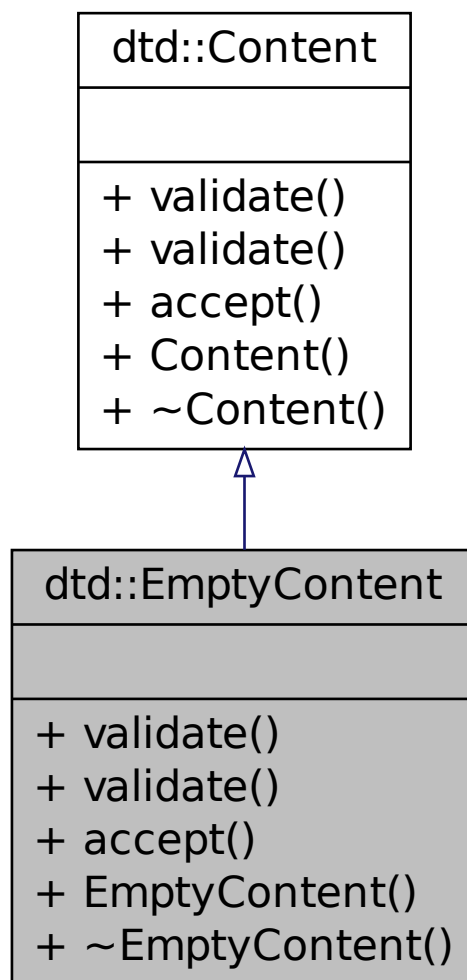
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/[ElementReference.hh](#)
- src/[ElementReference.cpp](#)

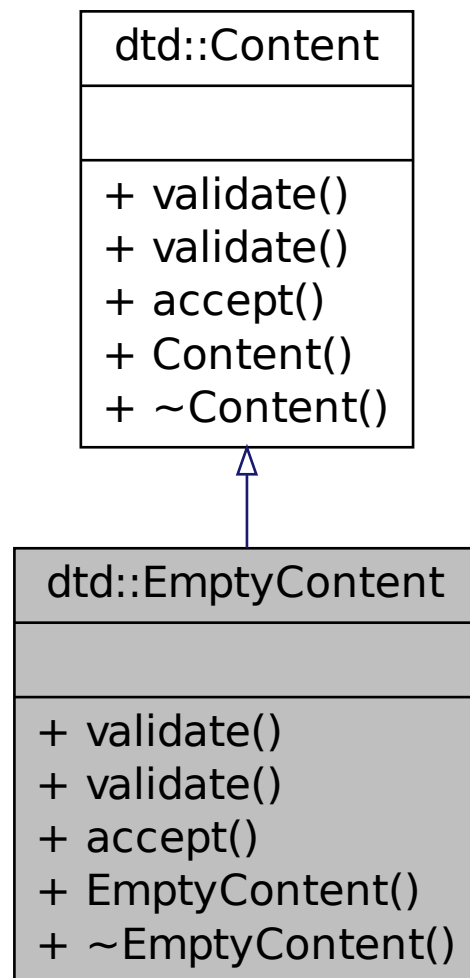
## 6.13 Référence de la classe dtd : :EmptyContent

```
#include <EmptyContent.hh>
```

Graphe d'héritage de dtd : :EmptyContent :



Graphe de collaboration de dtd : :EmptyContent :



### Fonctions membres publiques

- virtual bool `validate` (const xml : :MarkupNode &node)
- virtual bool `validate` (const xml : :CompositeMarkupNode &node)
- virtual void `accept` (InterfaceDTDVisitor &visitor) const
- `EmptyContent` ()
- virtual `~EmptyContent` ()

### 6.13.1 Description détaillée

Définition à la ligne 17 du fichier EmptyContent.hh.

### 6.13.2 Documentation des constructeurs et destructeur

#### 6.13.2.1 dtd : :EmptyContent : :EmptyContent ( )

Définition à la ligne 52 du fichier EmptyContent.cpp.

#### 6.13.2.2 dtd : :EmptyContent : :~EmptyContent ( ) [virtual]

Définition à la ligne 57 du fichier EmptyContent.cpp.

### 6.13.3 Documentation des fonctions membres

#### 6.13.3.1 void dtd : :EmptyContent : :accept ( InterfaceDTDVisitor & visitor ) const [virtual]

Implémente [dtd : :Content](#).

Définition à la ligne 43 du fichier EmptyContent.cpp.

Voici le graphe d'appel pour cette fonction :



#### 6.13.3.2 virtual bool dtd : :EmptyContent : :validate ( const xml : :MarkupNode & node ) [virtual]

Implémente [dtd : :Content](#).

#### 6.13.3.3 virtual bool dtd : :EmptyContent : :validate ( const xml : :CompositeMarkupNode & node ) [virtual]

Implémente [dtd : :Content](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

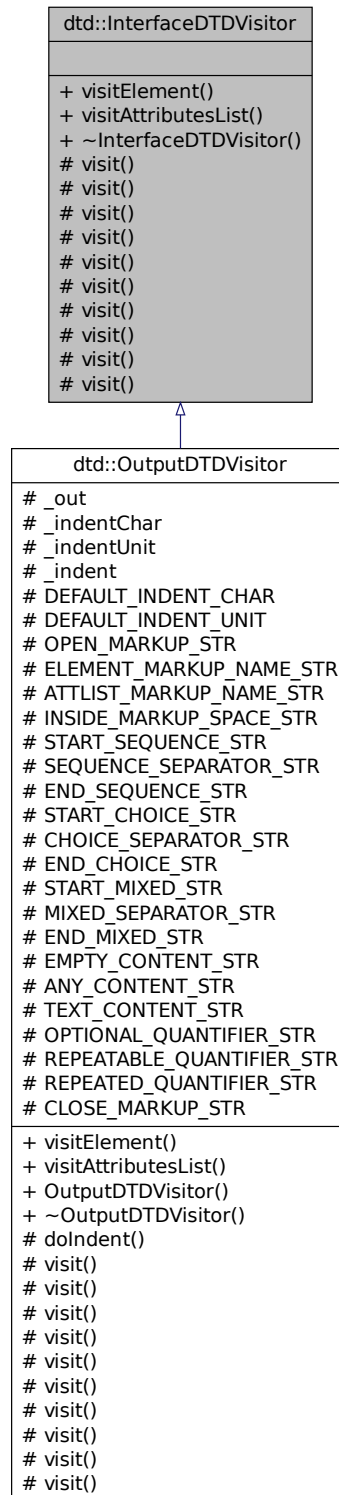
- [src/EmptyContent.hh](#)
- [src/EmptyContent.cpp](#)



## 6.14 Référence de la classe dtd : :InterfaceDTDVisitor

```
#include <InterfaceDTDVisitor.hpp>
```

Grappe d'héritage de dtd : :InterfaceDTDVisitor :



### Fonctions membres publiques

- virtual void [visitElement](#) (const std : :string &ns, const std : :string &elementName, const [Content](#) &content)=0
- virtual void [visitAttributesList](#) (const std : :string &ns, const std : :string &elementName, const [AttributesList](#) &attlist)=0
- virtual [~InterfaceDTDVisitor](#) ()

### Fonctions membres protégées

- virtual void [visit](#) (const [AnyContent](#) &content)=0
- virtual void [visit](#) (const [EmptyContent](#) &content)=0
- virtual void [visit](#) (const [MixedContent](#) &content)=0
- virtual void [visit](#) (const [TextContent](#) &content)=0
- virtual void [visit](#) (const [ElementReference](#) &element)=0
- virtual void [visit](#) (const [Choice](#) &content)=0
- virtual void [visit](#) (const [Sequence](#) &content)=0
- virtual void [visit](#) (const [OptionalContent](#) &content)=0
- virtual void [visit](#) (const [RepeatableContent](#) &content)=0
- virtual void [visit](#) (const [RepeatedContent](#) &content)=0

### Amis

- class [AnyContent](#)
- class [EmptyContent](#)
- class [MixedContent](#)
- class [TextContent](#)
- class [ElementReference](#)
- class [Choice](#)
- class [Sequence](#)
- class [OptionalContent](#)
- class [RepeatableContent](#)
- class [RepeatedContent](#)

#### 6.14.1 Description détaillée

Définition à la ligne 30 du fichier InterfaceDTDVisitor.hpp.

#### 6.14.2 Documentation des constructeurs et destructeur

**6.14.2.1** virtual dtd : :InterfaceDTDVisitor : :~InterfaceDTDVisitor ( ) [inline, virtual]

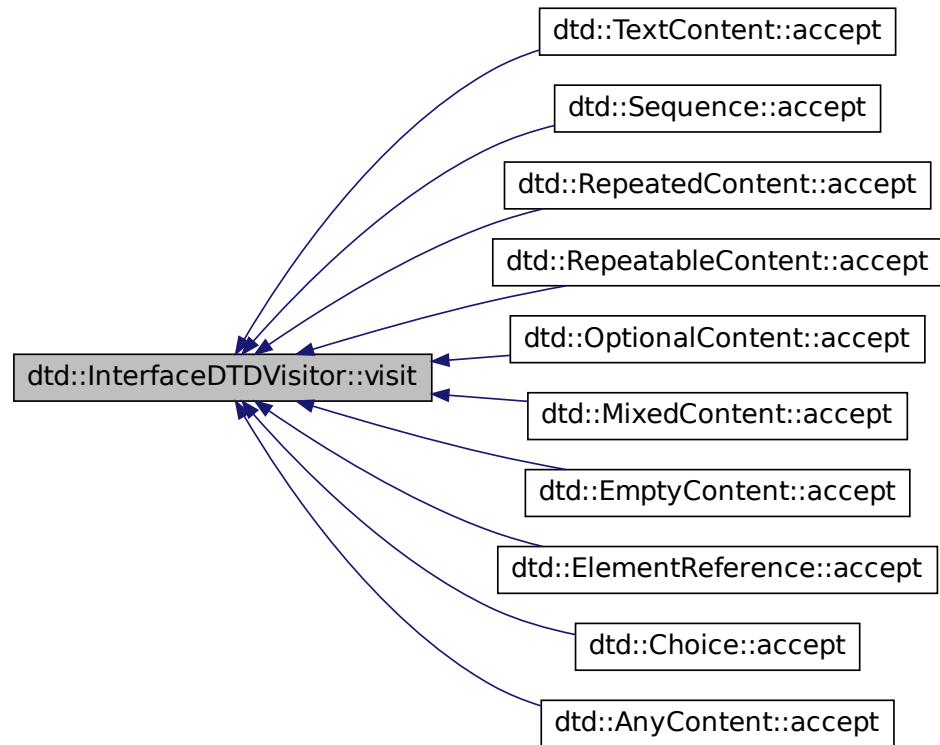
Définition à la ligne 46 du fichier InterfaceDTDVisitor.hpp.

#### 6.14.3 Documentation des fonctions membres

**6.14.3.1** virtual void dtd : :InterfaceDTDVisitor : :visit ( const AnyContent & *content* ) [protected, pure virtual]

Implémenté dans [dtd : :OutputDTDVisitor](#).

Voici le graphe d'appel pour cette fonction :



**6.14.3.2** `virtual void dtd : :InterfaceDTDVisitor : :visit ( const EmptyContent & content )`  
`[protected, pure virtual]`

Implémenté dans [dtd : :OutputDTDVisitor](#).

**6.14.3.3** `virtual void dtd : :InterfaceDTDVisitor : :visit ( const RepeatableContent & content )`  
`[protected, pure virtual]`

Implémenté dans [dtd : :OutputDTDVisitor](#).

**6.14.3.4** `virtual void dtd : :InterfaceDTDVisitor : :visit ( const OptionalContent & content )`  
`[protected, pure virtual]`

Implémenté dans [dtd : :OutputDTDVisitor](#).

**6.14.3.5** `virtual void dtd : :InterfaceDTDVisitor : :visit ( const Choice & content )`  
[protected, pure virtual]

Implémenté dans [dtd : :OutputDTDVisitor](#).

**6.14.3.6** `virtual void dtd : :InterfaceDTDVisitor : :visit ( const Sequence & content )`  
[protected, pure virtual]

Implémenté dans [dtd : :OutputDTDVisitor](#).

**6.14.3.7** `virtual void dtd : :InterfaceDTDVisitor : :visit ( const ElementReference & element )`  
[protected, pure virtual]

Implémenté dans [dtd : :OutputDTDVisitor](#).

**6.14.3.8** `virtual void dtd : :InterfaceDTDVisitor : :visit ( const RepeatedContent & content )`  
[protected, pure virtual]

Implémenté dans [dtd : :OutputDTDVisitor](#).

**6.14.3.9** `virtual void dtd : :InterfaceDTDVisitor : :visit ( const MixedContent & content )`  
[protected, pure virtual]

Implémenté dans [dtd : :OutputDTDVisitor](#).

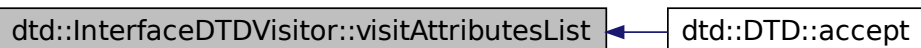
**6.14.3.10** `virtual void dtd : :InterfaceDTDVisitor : :visit ( const TextContent & content )`  
[protected, pure virtual]

Implémenté dans [dtd : :OutputDTDVisitor](#).

**6.14.3.11** `virtual void dtd : :InterfaceDTDVisitor : :visitAttributesList ( const std : :string & ns, const std : :string & elementName, const AttributesList & attlist )` [pure virtual]

Implémenté dans [dtd : :OutputDTDVisitor](#).

Voici le graphe d'appel pour cette fonction :



**6.14.3.12** `virtual void dtd : :InterfaceDTDVisitor : :visitElement ( const std : :string & ns, const std : :string & elementName, const Content & content ) [pure virtual]`

Implémenté dans [dtd : :OutputDTDVisitor](#).

Voici le graphe d'appel pour cette fonction :



#### 6.14.4 Documentation des fonctions amies et associées

**6.14.4.1** `friend class AnyContent [friend]`

Définition à la ligne 56 du fichier `InterfaceDTDVisitor.hpp`.

**6.14.4.2** `friend class Choice [friend]`

Définition à la ligne 61 du fichier `InterfaceDTDVisitor.hpp`.

**6.14.4.3** `friend class ElementReference [friend]`

Définition à la ligne 60 du fichier `InterfaceDTDVisitor.hpp`.

**6.14.4.4** `friend class EmptyContent [friend]`

Définition à la ligne 57 du fichier `InterfaceDTDVisitor.hpp`.

**6.14.4.5** `friend class MixedContent [friend]`

Définition à la ligne 58 du fichier `InterfaceDTDVisitor.hpp`.

**6.14.4.6** `friend class OptionalContent [friend]`

Définition à la ligne 63 du fichier `InterfaceDTDVisitor.hpp`.

**6.14.4.7** `friend class RepeatableContent [friend]`

Définition à la ligne 64 du fichier `InterfaceDTDVisitor.hpp`.

#### 6.14.4.8 friend class RepeatedContent [friend]

Définition à la ligne 65 du fichier InterfaceDTDVisitor.hpp.

#### 6.14.4.9 friend class Sequence [friend]

Définition à la ligne 62 du fichier InterfaceDTDVisitor.hpp.

#### 6.14.4.10 friend class TextContent [friend]

Définition à la ligne 59 du fichier InterfaceDTDVisitor.hpp.

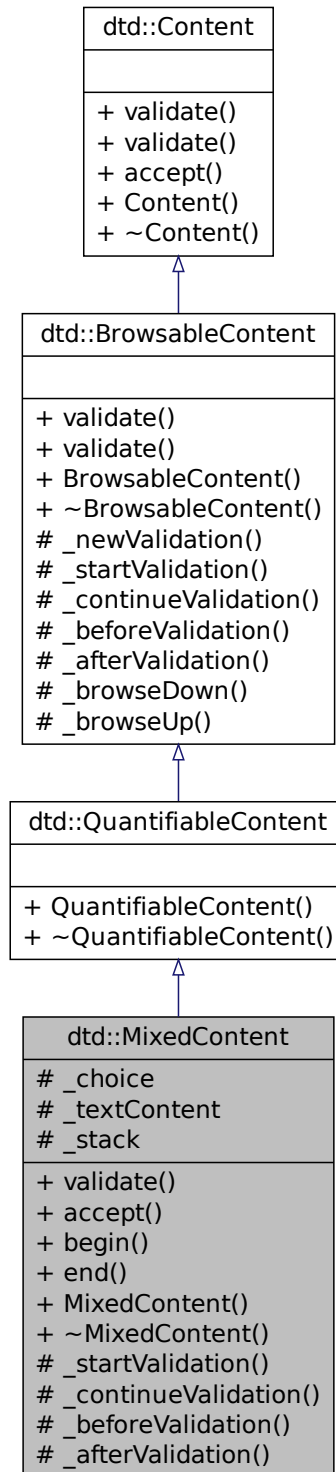
La documentation de cette classe a été générée à partir du fichier suivant :

— src/[InterfaceDTDVisitor.hpp](#)

## 6.15 Référence de la classe dtd : :MixedContent

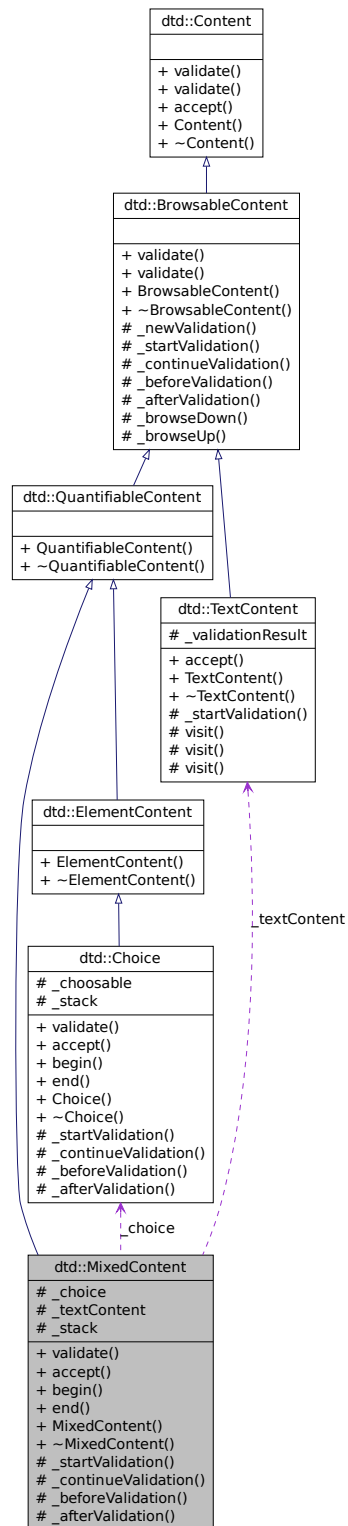
```
#include <MixedContent.hh>
```

Graphe d'héritage de dtd : :MixedContent :





Grphe de collaboration de dtd : :MixedContent :



## Types publics

- typedef [Choice](#) : [:ChoosableSet](#) [ChoosableSet](#)
- typedef [ChoosableSet](#) : [:const\\_iterator](#) [const\\_iterator](#)

## Fonctions membres publiques

- virtual bool [validate](#) (const xml : [:CompositeMarkupNode](#) &node)
- virtual void [accept](#) ([InterfaceDTDVisitor](#) &visitor) const
- [const\\_iterator begin](#) () const
- [const\\_iterator end](#) () const
- [MixedContent](#) ([TextContent](#) &textContent, const [ChoosableSet](#) &choosable)
- virtual [~MixedContent](#) ()

## Types protégés

- typedef std : [:stack<\\_State>](#) [\\_StatesStack](#)

## Fonctions membres protégées

- virtual bool [\\_startValidation](#) (xml : [:CompositeMarkupNode](#) : [:ChildrenIterator](#) firstToken, xml : [:CompositeMarkupNode](#) : [:ChildrenIterator](#) endToken, [BrowsableContent](#) \*nextStep)
- virtual bool [\\_continueValidation](#) (xml : [:CompositeMarkupNode](#) : [:ChildrenIterator](#) currentToken)
- virtual void [\\_beforeValidation](#) (xml : [:CompositeMarkupNode](#) : [:ChildrenIterator](#) firstToken, xml : [:CompositeMarkupNode](#) : [:ChildrenIterator](#) endToken, [BrowsableContent](#) \*nextStep)
- virtual void [\\_afterValidation](#) ()

## Attributs protégés

- [Choice](#) [\\_choice](#)
- [TextContent](#) & [\\_textContent](#)
- [\\_StatesStack](#) [\\_stack](#)

### 6.15.1 Description détaillée

Définition à la ligne 21 du fichier MixedContent.hh.

### 6.15.2 Documentation des définitions de type membres

#### 6.15.2.1 typedef std : [:stack<\\_State>](#) dtd : [:MixedContent](#) : [\\_StatesStack](#) [protected]

Définition à la ligne 84 du fichier MixedContent.hh.

#### 6.15.2.2 typedef Choice : :ChoosableSet dtd : :MixedContent : :ChoosableSet

Définition à la ligne 29 du fichier MixedContent.hh.

#### 6.15.2.3 typedef ChoosableSet : :const\_iterator dtd : :MixedContent : :const\_iterator

Définition à la ligne 30 du fichier MixedContent.hh.

### 6.15.3 Documentation des constructeurs et destructeur

#### 6.15.3.1 dtd : :MixedContent : :MixedContent ( TextContent & *textContent*, const ChoosableSet & *choosable* )

Définition à la ligne 61 du fichier MixedContent.cpp.

#### 6.15.3.2 dtd : :MixedContent : :~MixedContent ( ) [virtual]

Définition à la ligne 68 du fichier MixedContent.cpp.

### 6.15.4 Documentation des fonctions membres

#### 6.15.4.1 void dtd : :MixedContent : :afterValidation ( ) [protected, virtual]

Réimplémentée à partir de [dtd : :BrowsableContent](#).

Définition à la ligne 84 du fichier MixedContent.cpp.

#### 6.15.4.2 void dtd : :MixedContent : :beforeValidation ( xml : :CompositeMarkupNode : :ChildrenIterator *firstToken*, xml : :CompositeMarkupNode : :ChildrenIterator *endToken*, BrowsableContent \* *nextStep* ) [protected, virtual]

Réimplémentée à partir de [dtd : :BrowsableContent](#).

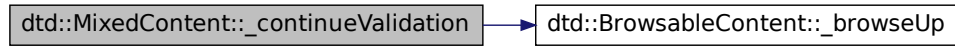
Définition à la ligne 76 du fichier MixedContent.cpp.

#### 6.15.4.3 bool dtd : :MixedContent : :continueValidation ( xml : :CompositeMarkupNode : :ChildrenIterator *currentToken* ) [protected, virtual]

Réimplémentée à partir de [dtd : :BrowsableContent](#).

Définition à la ligne 97 du fichier MixedContent.cpp.

Voici le graphe d'appel pour cette fonction :



**6.15.4.4** `virtual bool dtd : :MixedContent : :_startValidation (`  
`xml : :CompositeMarkupNode : :ChildrenIterator firstToken,`  
`xml : :CompositeMarkupNode : :ChildrenIterator endToken, BrowsableContent *`  
`nextStep ) [protected, virtual]`

Implémente [dtd : :BrowsableContent](#).

**6.15.4.5** `void dtd : :MixedContent : :accept ( InterfaceDTDVisitor & visitor ) const`  
`[virtual]`

Implémente [dtd : :Content](#).

Définition à la ligne 42 du fichier MixedContent.cpp.

Voici le graphe d'appel pour cette fonction :



**6.15.4.6** `MixedContent : :const_iterator dtd : :MixedContent : :begin ( ) const`

Définition à la ligne 47 du fichier MixedContent.cpp.

**6.15.4.7** `MixedContent : :const_iterator dtd : :MixedContent : :end ( ) const`

Définition à la ligne 52 du fichier MixedContent.cpp.

**6.15.4.8** `virtual bool dtd : :MixedContent : :validate ( const xml : :CompositeMarkupNode &`  
`node ) [virtual]`

Réimplémentée à partir de [dtd : :BrowsableContent](#).

### 6.15.5 Documentation des données membres

#### 6.15.5.1 Choice dtd : :MixedContent : :\_choice [protected]

Définition à la ligne 81 du fichier MixedContent.hh.

#### 6.15.5.2 \_StatesStack dtd : :MixedContent : :\_stack [protected]

Définition à la ligne 85 du fichier MixedContent.hh.

#### 6.15.5.3 TextContent& dtd : :MixedContent : :\_textContent [protected]

Définition à la ligne 82 du fichier MixedContent.hh.

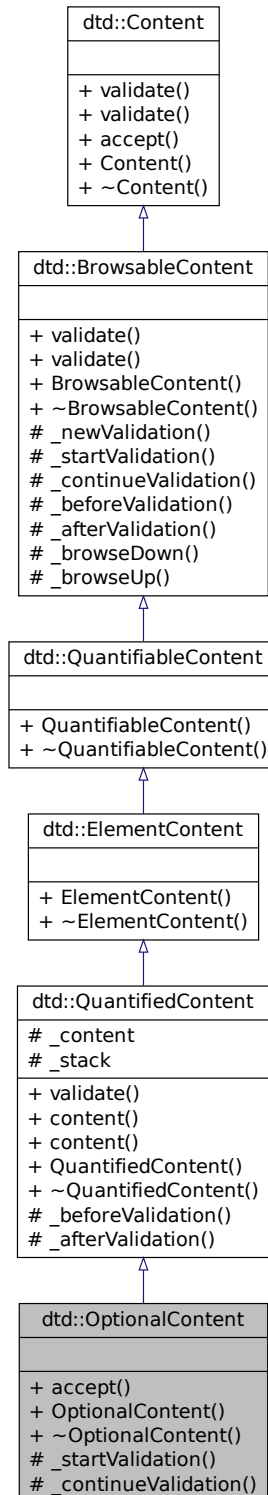
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/[MixedContent.hh](#)
- src/[MixedContent.cpp](#)

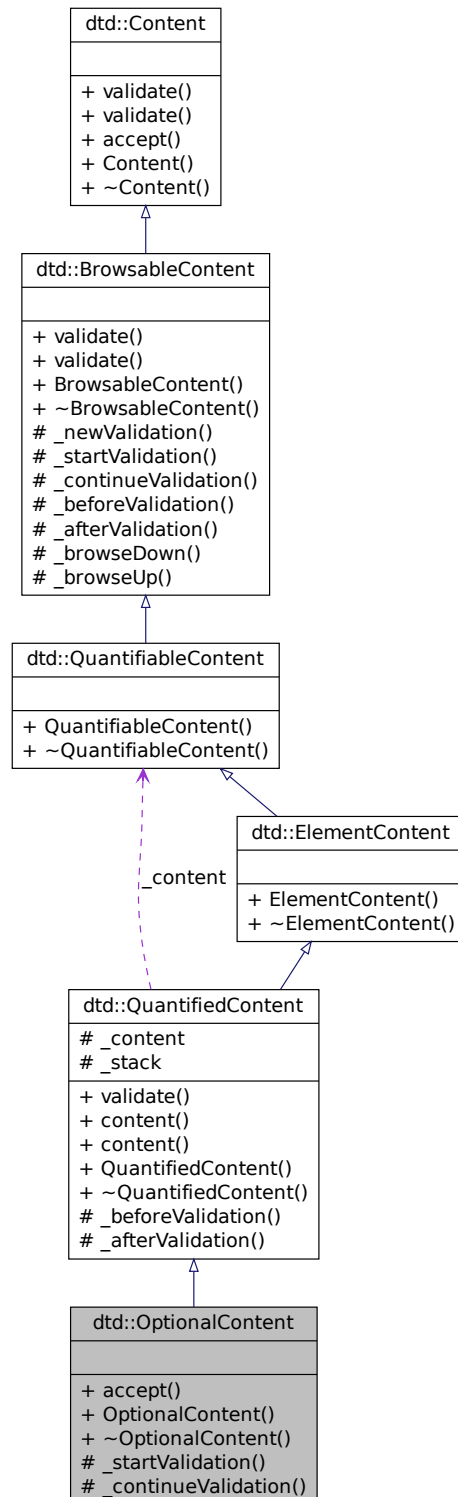
## 6.16 Référence de la classe dtd : :OptionalContent

```
#include <OptionalContent.hh>
```

Graphe d'héritage de dtd : :OptionalContent :



Graphe de collaboration de dtd : :OptionalContent :



### Fonctions membres publiques

- virtual void [accept](#) ([InterfaceDTDVisitor](#) &visitor) const
- [OptionalContent](#) ([QuantifiableContent](#) &content)
- virtual [~OptionalContent](#) ()

### Fonctions membres protégées

- virtual bool [\\_startValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- virtual bool [\\_continueValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator currentToken)

#### 6.16.1 Description détaillée

Définition à la ligne 18 du fichier OptionalContent.hh.

#### 6.16.2 Documentation des constructeurs et destructeur

##### 6.16.2.1 dtd : :OptionalContent : :OptionalContent ( [QuantifiableContent](#) & *content* )

Définition à la ligne 42 du fichier OptionalContent.cpp.

##### 6.16.2.2 dtd : :OptionalContent : :~OptionalContent ( ) [virtual]

Définition à la ligne 48 du fichier OptionalContent.cpp.

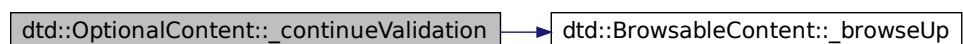
#### 6.16.3 Documentation des fonctions membres

##### 6.16.3.1 bool dtd : :OptionalContent : :\_continueValidation ( xml : :CompositeMarkupNode : :ChildrenIterator *currentToken* ) [protected, virtual]

Réimplémentée à partir de [dtd : :BrowsableContent](#).

Définition à la ligne 78 du fichier OptionalContent.cpp.

Voici le graphe d'appel pour cette fonction :





```
6.16.3.2 virtual bool dtd : :OptionalContent : :startValidation (
    xml : :CompositeMarkupNode : :ChildrenIterator firstToken,
    xml : :CompositeMarkupNode : :ChildrenIterator endToken, BrowsableContent *
    nextStep ) [protected, virtual]
```

Implémente [dtd : :BrowsableContent](#).

```
6.16.3.3 void dtd : :OptionalContent : :accept ( InterfaceDTDVisitor & visitor ) const
    [virtual]
```

Implémente [dtd : :Content](#).

Définition à la ligne 33 du fichier OptionalContent.cpp.

Voici le graphe d'appel pour cette fonction :



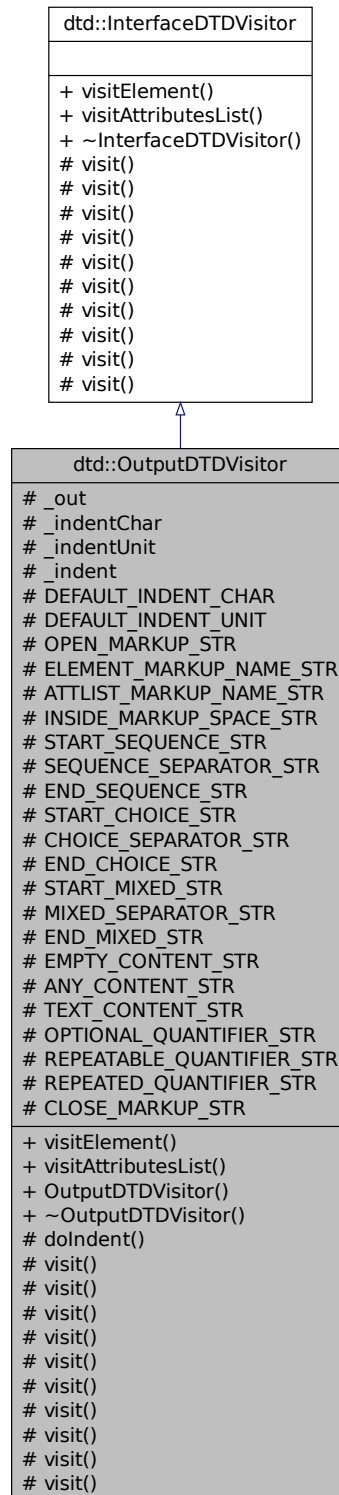
La documentation de cette classe a été générée à partir des fichiers suivants :

- [src/OptionalContent.hh](#)
- [src/OptionalContent.cpp](#)

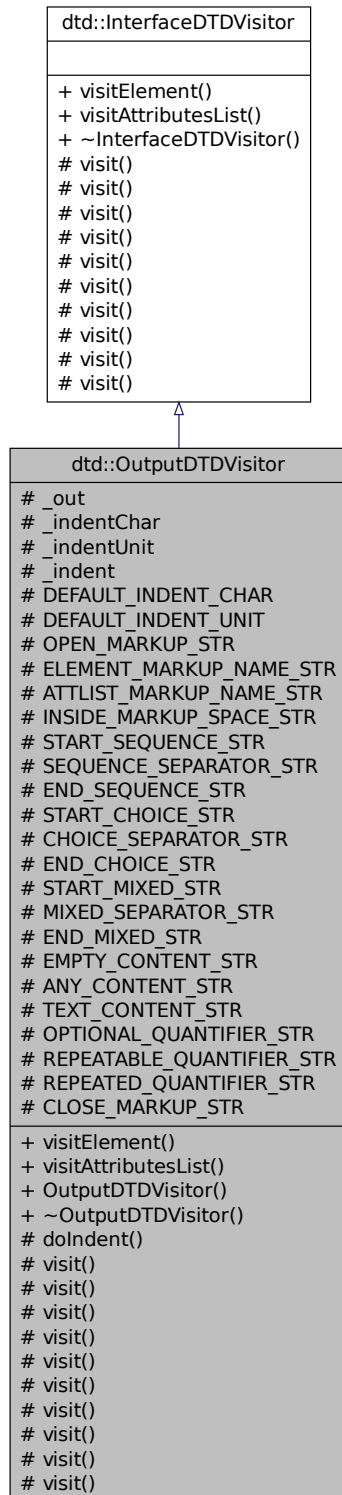
## 6.17 Référence de la classe dtd : :OutputDTDVisitor

```
#include <OutputDTDVisitor.hh>
```

Graphe d'héritage de dtd : :OutputDTDVisitor :



Graphe de collaboration de dtd : :OutputDTDVisitor :



## Fonctions membres publiques

- virtual void `visitElement` (const std : :string &ns, const std : :string &elementName, const `Content` &content)
- virtual void `visitAttributesList` (const std : :string &ns, const std : :string &elementName, const `AttributesList` &attlist)
- `OutputDTDVisitor` (std : :ostream &out, char indentChar=`DEFAULT_INDENT_CHAR`, unsigned int indentUnit=`DEFAULT_INDENT_UNIT`)
- virtual `~OutputDTDVisitor` ()

## Fonctions membres protégées

- void `doIndent` ()
- virtual void `visit` (const `EmptyContent` &content)
- virtual void `visit` (const `AnyContent` &content)
- virtual void `visit` (const `MixedContent` &content)
- virtual void `visit` (const `TextContent` &content)
- virtual void `visit` (const `ElementReference` &element)
- virtual void `visit` (const `Choice` &content)
- virtual void `visit` (const `Sequence` &content)
- virtual void `visit` (const `OptionalContent` &content)
- virtual void `visit` (const `RepeatableContent` &content)
- virtual void `visit` (const `RepeatedContent` &content)

## Attributs protégés

- std : :ostream & `_out`
- char `_indentChar`
- unsigned `_indentUnit`
- unsigned int `_indent`

## Attributs protégés statiques

- static const char `DEFAULT_INDENT_CHAR` = '\t'
- static const unsigned char `DEFAULT_INDENT_UNIT` = 1
- static const std : :string `OPEN_MARKUP_STR` = "<!"
- static const std : :string `ELEMENT_MARKUP_NAME_STR` = "ELEMENT"
- static const std : :string `ATTLIST_MARKUP_NAME_STR` = "ATTLIST"
- static const std : :string `INSIDE_MARKUP_SPACE_STR` = " "
- static const std : :string `START_SEQUENCE_STR` = "("
- static const std : :string `SEQUENCE_SEPARATOR_STR` = ","
- static const std : :string `END_SEQUENCE_STR` = ")"
- static const std : :string `START_CHOICE_STR` = "("
- static const std : :string `CHOICE_SEPARATOR_STR` = "|"
- static const std : :string `END_CHOICE_STR` = ")"
- static const std : :string `START_MIXED_STR` = "("
- static const std : :string `MIXED_SEPARATOR_STR` = "|"
- static const std : :string `END_MIXED_STR` = ")"
- static const std : :string `EMPTY_CONTENT_STR` = "EMPTY"
- static const std : :string `ANY_CONTENT_STR` = "ANY"
- static const std : :string `TEXT_CONTENT_STR` = "#PCDATA"
- static const std : :string `OPTIONAL_QUANTIFIER_STR` = "?"
- static const std : :string `REPEATABLE_QUANTIFIER_STR` = "\*"
- static const std : :string `REPEATED_QUANTIFIER_STR` = "+"
- static const std : :string `CLOSE_MARKUP_STR` = ">"

### 6.17.1 Description détaillée

Définition à la ligne 20 du fichier `OutputDTDVisitor.hh`.

### 6.17.2 Documentation des constructeurs et destructeur

**6.17.2.1** `dtd : :OutputDTDVisitor ( std : :ostream & out, char indentChar = DEFAULT_INDENT_CHAR, unsigned int indentUnit = DEFAULT_INDENT_UNIT )`

Définition à la ligne 94 du fichier `OutputDTDVisitor.cpp`.

**6.17.2.2** `dtd : :OutputDTDVisitor : :~OutputDTDVisitor ( ) [virtual]`

Définition à la ligne 101 du fichier `OutputDTDVisitor.cpp`.

### 6.17.3 Documentation des fonctions membres

**6.17.3.1** `void dtd : :OutputDTDVisitor : :doIndent ( ) [protected]`

Définition à la ligne 110 du fichier `OutputDTDVisitor.cpp`.

**6.17.3.2** `void dtd : :OutputDTDVisitor : :visit ( const RepeatedContent & content ) [protected, virtual]`

Implémente [dtd : :InterfaceDTDVisitor](#).

Définition à la ligne 195 du fichier `OutputDTDVisitor.cpp`.

**6.17.3.3** `void dtd : :OutputDTDVisitor : :visit ( const EmptyContent & content ) [protected, virtual]`

Implémente [dtd : :InterfaceDTDVisitor](#).

Définition à la ligne 125 du fichier `OutputDTDVisitor.cpp`.

**6.17.3.4** `void dtd : :OutputDTDVisitor : :visit ( const OptionalContent & content ) [protected, virtual]`

Implémente [dtd : :InterfaceDTDVisitor](#).

Définition à la ligne 183 du fichier `OutputDTDVisitor.cpp`.

**6.17.3.5** `void dtd : :OutputDTDVisitor : :visit ( const Sequence & content )` [protected, virtual]

Implémente [dtd : :InterfaceDTDVisitor](#).

Définition à la ligne 168 du fichier OutputDTDVisitor.cpp.

**6.17.3.6** `void dtd : :OutputDTDVisitor : :visit ( const ElementReference & element )` [protected, virtual]

Implémente [dtd : :InterfaceDTDVisitor](#).

Définition à la ligne 148 du fichier OutputDTDVisitor.cpp.

**6.17.3.7** `void dtd : :OutputDTDVisitor : :visit ( const Choice & content )` [protected, virtual]

Implémente [dtd : :InterfaceDTDVisitor](#).

Définition à la ligne 153 du fichier OutputDTDVisitor.cpp.

**6.17.3.8** `void dtd : :OutputDTDVisitor : :visit ( const TextContent & content )` [protected, virtual]

Implémente [dtd : :InterfaceDTDVisitor](#).

Définition à la ligne 143 du fichier OutputDTDVisitor.cpp.

**6.17.3.9** `void dtd : :OutputDTDVisitor : :visit ( const RepeatableContent & content )` [protected, virtual]

Implémente [dtd : :InterfaceDTDVisitor](#).

Définition à la ligne 189 du fichier OutputDTDVisitor.cpp.

**6.17.3.10** `void dtd : :OutputDTDVisitor : :visit ( const AnyContent & content )` [protected, virtual]

Implémente [dtd : :InterfaceDTDVisitor](#).

Définition à la ligne 120 du fichier OutputDTDVisitor.cpp.

**6.17.3.11** `void dtd : :OutputDTDVisitor : :visit ( const MixedContent & content )` [protected, virtual]

Implémente [dtd : :InterfaceDTDVisitor](#).

Définition à la ligne 130 du fichier OutputDTDVisitor.cpp.

**6.17.3.12** void dtd : :OutputDTDVisitor : :visitAttributesList ( const std : :string & ns, const std : :string & elementName, const AttributesList & attlist ) [virtual]

Implémente [dtd : :InterfaceDTDVisitor](#).

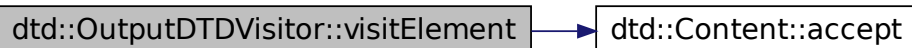
Définition à la ligne 70 du fichier OutputDTDVisitor.cpp.

**6.17.3.13** void dtd : :OutputDTDVisitor : :visitElement ( const std : :string & ns, const std : :string & elementName, const Content & content ) [virtual]

Implémente [dtd : :InterfaceDTDVisitor](#).

Définition à la ligne 60 du fichier OutputDTDVisitor.cpp.

Voici le graphe d'appel pour cette fonction :



## 6.17.4 Documentation des données membres

**6.17.4.1** unsigned int dtd : :OutputDTDVisitor : :\_indent [protected]

Définition à la ligne 79 du fichier OutputDTDVisitor.hh.

**6.17.4.2** char dtd : :OutputDTDVisitor : :\_indentChar [protected]

Définition à la ligne 77 du fichier OutputDTDVisitor.hh.

**6.17.4.3** unsigned dtd : :OutputDTDVisitor : :\_indentUnit [protected]

Définition à la ligne 78 du fichier OutputDTDVisitor.hh.

**6.17.4.4** std : :ostream& dtd : :OutputDTDVisitor : :\_out [protected]

Définition à la ligne 76 du fichier OutputDTDVisitor.hh.

**6.17.4.5** const std : :string dtd : :OutputDTDVisitor : :ANY\_CONTENT\_STR = "ANY"  
[static, protected]

Définition à la ligne 69 du fichier OutputDTDVisitor.hh.

**6.17.4.6** `const std::string dtd : :OutputDTDVisitor : :ATTLIST_MARKUP_NAME_STR =  
"ATTLIST" [static, protected]`

Définition à la ligne 57 du fichier OutputDTDVisitor.hh.

**6.17.4.7** `const std::string dtd : :OutputDTDVisitor : :CHOICE_SEPARATOR_STR = "|"  
[static, protected]`

Définition à la ligne 63 du fichier OutputDTDVisitor.hh.

**6.17.4.8** `const std::string dtd : :OutputDTDVisitor : :CLOSE_MARKUP_STR = ">"  
[static, protected]`

Définition à la ligne 74 du fichier OutputDTDVisitor.hh.

**6.17.4.9** `const char dtd : :OutputDTDVisitor : :DEFAULT_INDENT_CHAR = '\t' [static,  
protected]`

Définition à la ligne 23 du fichier OutputDTDVisitor.hh.

**6.17.4.10** `const unsigned char dtd : :OutputDTDVisitor : :DEFAULT_INDENT_UNIT = 1  
[static, protected]`

Définition à la ligne 24 du fichier OutputDTDVisitor.hh.

**6.17.4.11** `const std::string dtd : :OutputDTDVisitor : :ELEMENT_MARKUP_NAME_STR =  
"ELEMENT" [static, protected]`

Définition à la ligne 56 du fichier OutputDTDVisitor.hh.

**6.17.4.12** `const std::string dtd : :OutputDTDVisitor : :EMPTY_CONTENT_STR = "EMPTY"  
[static, protected]`

Définition à la ligne 68 du fichier OutputDTDVisitor.hh.

**6.17.4.13** `const std::string dtd : :OutputDTDVisitor : :END_CHOICE_STR = ")" [static,  
protected]`

Définition à la ligne 64 du fichier OutputDTDVisitor.hh.

**6.17.4.14** `const std::string dtd : :OutputDTDVisitor : :END_MIXED_STR = ")" [static,  
protected]`

Définition à la ligne 67 du fichier OutputDTDVisitor.hh.



**6.17.4.15** `const std::string dtd::OutputDTDVisitor::END_SEQUENCE_STR = ")"`  
`[static, protected]`

Définition à la ligne 61 du fichier OutputDTDVisitor.hh.

**6.17.4.16** `const std::string dtd::OutputDTDVisitor::INSIDE_MARKUP_SPACE_STR = " "`  
`[static, protected]`

Définition à la ligne 58 du fichier OutputDTDVisitor.hh.

**6.17.4.17** `const std::string dtd::OutputDTDVisitor::MIXED_SEPARATOR_STR = "|"`  
`[static, protected]`

Définition à la ligne 66 du fichier OutputDTDVisitor.hh.

**6.17.4.18** `const std::string dtd::OutputDTDVisitor::OPEN_MARKUP_STR = "<!"`  
`[static, protected]`

Définition à la ligne 55 du fichier OutputDTDVisitor.hh.

**6.17.4.19** `const std::string dtd::OutputDTDVisitor::OPTIONAL_QUANTIFIER_STR = "?"`  
`[static, protected]`

Définition à la ligne 71 du fichier OutputDTDVisitor.hh.

**6.17.4.20** `const std::string dtd::OutputDTDVisitor::REPEATABLE_QUANTIFIER_STR = "*"`  
`[static, protected]`

Définition à la ligne 72 du fichier OutputDTDVisitor.hh.

**6.17.4.21** `const std::string dtd::OutputDTDVisitor::REPEATED_QUANTIFIER_STR = "+"`  
`[static, protected]`

Définition à la ligne 73 du fichier OutputDTDVisitor.hh.

**6.17.4.22** `const std::string dtd::OutputDTDVisitor::SEQUENCE_SEPARATOR_STR = ";"`  
`[static, protected]`

Définition à la ligne 60 du fichier OutputDTDVisitor.hh.

**6.17.4.23** `const std::string dtd::OutputDTDVisitor::START_CHOICE_STR = "("` `[static, protected]`

Définition à la ligne 62 du fichier OutputDTDVisitor.hh.

**6.17.4.24** `const std::string dtd::OutputDTDVisitor::START_MIXED_STR = "("` [static, protected]

Définition à la ligne 65 du fichier OutputDTDVisitor.hh.

**6.17.4.25** `const std::string dtd::OutputDTDVisitor::START_SEQUENCE_STR = "("` [static, protected]

Définition à la ligne 59 du fichier OutputDTDVisitor.hh.

**6.17.4.26** `const std::string dtd::OutputDTDVisitor::TEXT_CONTENT_STR = "#PCDATA"` [static, protected]

Définition à la ligne 70 du fichier OutputDTDVisitor.hh.

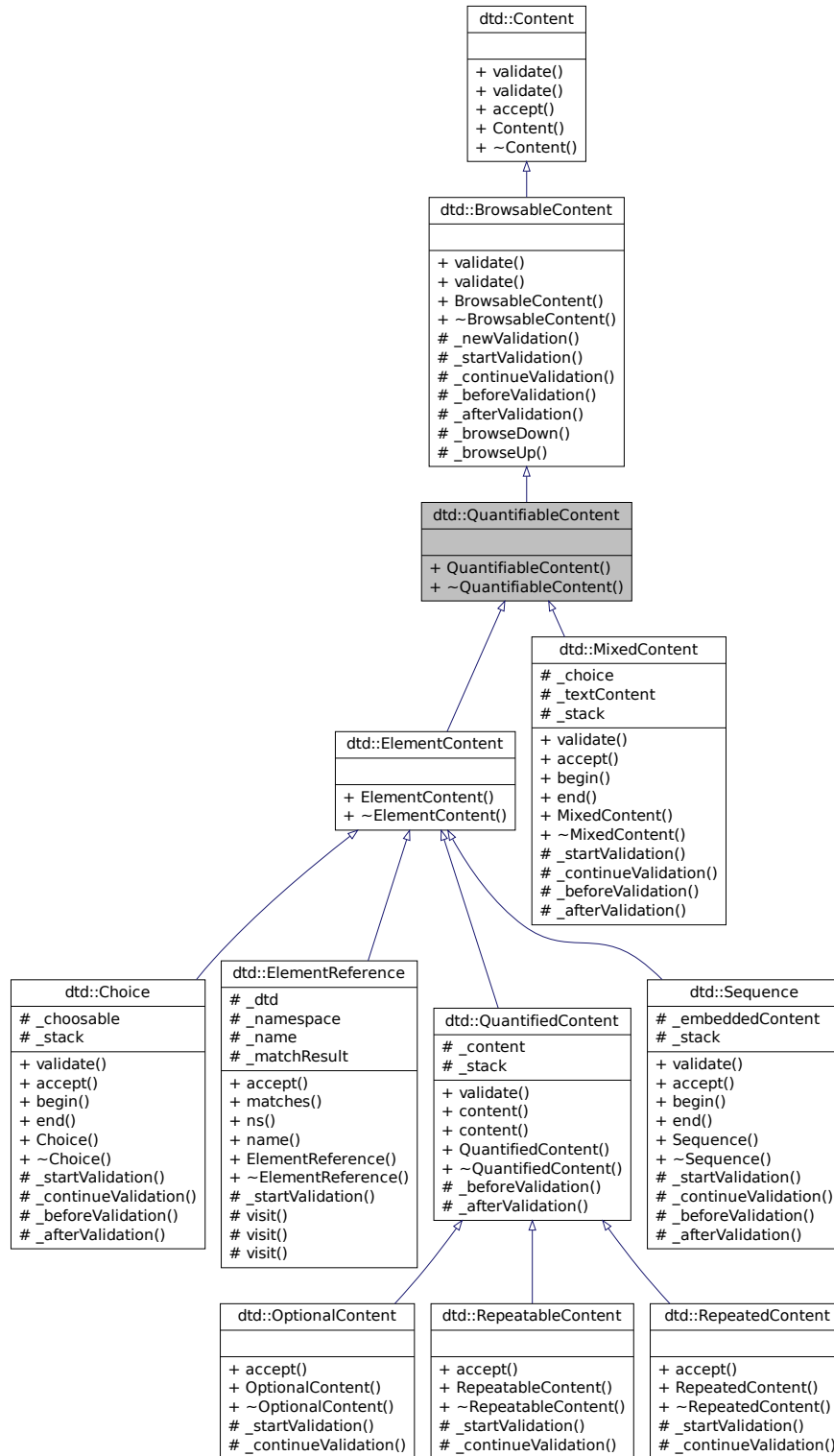
La documentation de cette classe a été générée à partir des fichiers suivants :

- [src/OutputDTDVisitor.hh](#)
- [src/OutputDTDVisitor.cpp](#)

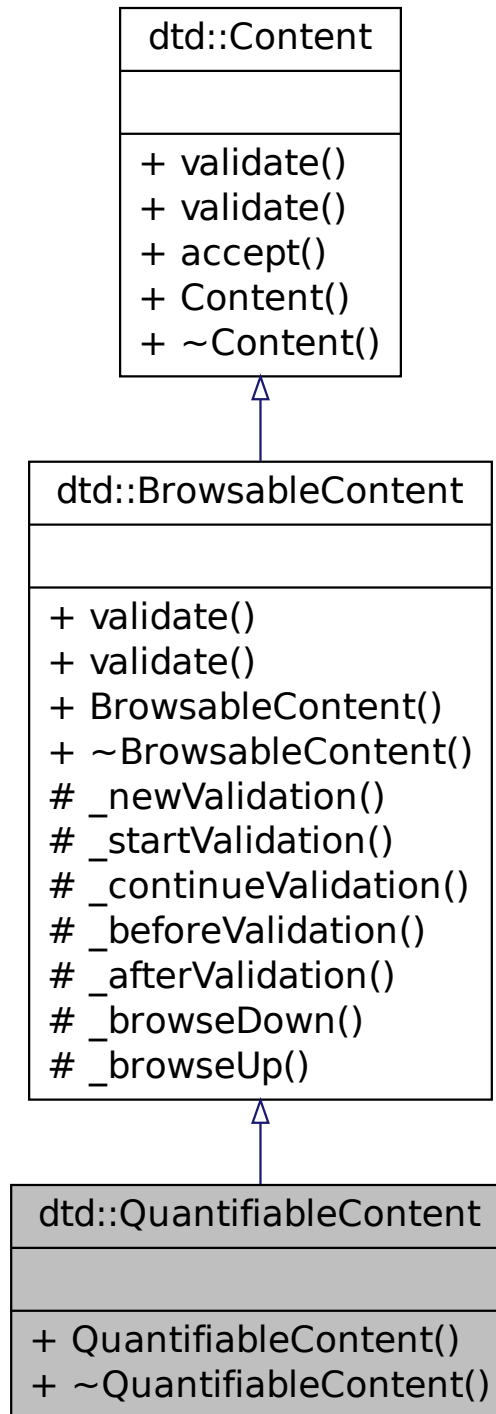
## 6.18 Référence de la classe dtd::QuantifiableContent

```
#include <QuantifiableContent.hh>
```

Graphe d'héritage de dtd : :QuantifiableContent :



Graphe de collaboration de dtd::QuantifiableContent :



## Classes

- struct [\\_State](#)

## Fonctions membres publiques

- [QuantifiableContent](#) ()
- virtual [~QuantifiableContent](#) ()

### 6.18.1 Description détaillée

Définition à la ligne 18 du fichier QuantifiableContent.hh.

### 6.18.2 Documentation des constructeurs et destructeur

#### 6.18.2.1 dtd : :QuantifiableContent : :QuantifiableContent ( )

Définition à la ligne 36 du fichier QuantifiableContent.cpp.

#### 6.18.2.2 dtd : :QuantifiableContent : :~QuantifiableContent ( ) [virtual]

Définition à la ligne 41 du fichier QuantifiableContent.cpp.

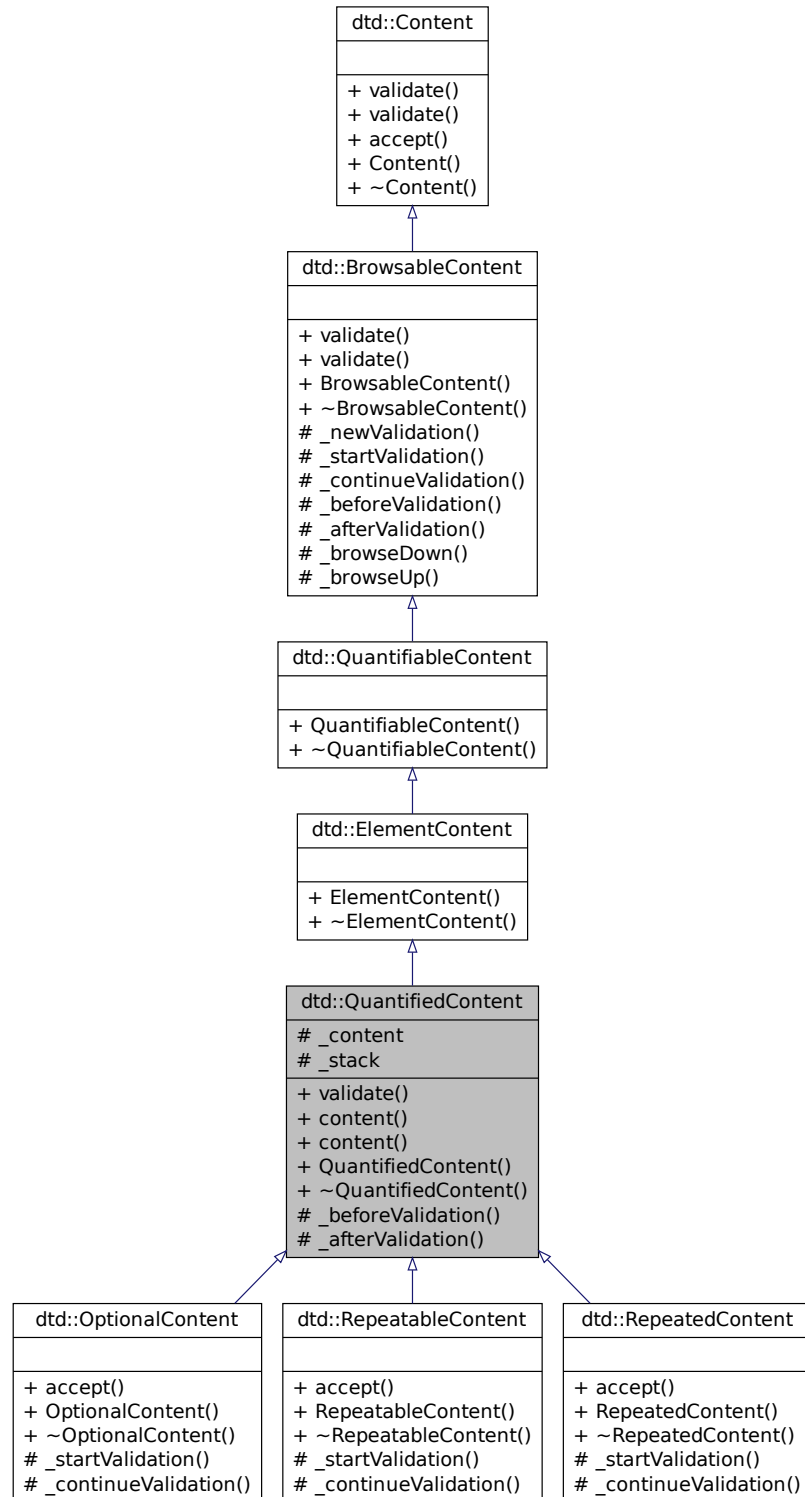
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/[QuantifiableContent.hh](#)
- src/[QuantifiableContent.cpp](#)

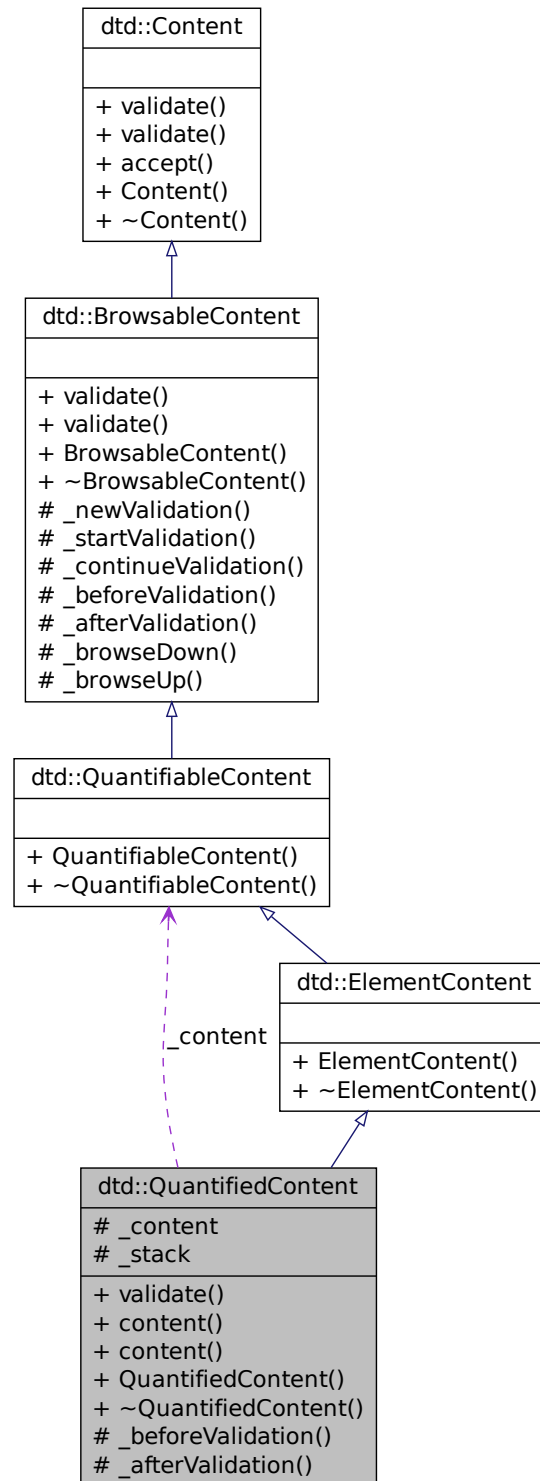
## 6.19 Référence de la classe dtd : :QuantifiedContent

```
#include <QuantifiedContent.hh>
```

Graphe d'héritage de dtd : :QuantifiedContent :



Grphe de collaboration de dtd : :QuantifiedContent :



### Fonctions membres publiques

- virtual bool [validate](#) (const xml : :CompositeMarkupNode &node)
- [QuantifiableContent](#) & [content](#) ()
- const [QuantifiableContent](#) & [content](#) () const
- [QuantifiedContent](#) ([QuantifiableContent](#) &content)
- virtual [~QuantifiedContent](#) ()

### Types protégés

- typedef std : :stack< [\\_State](#) > [\\_StatesStack](#)

### Fonctions membres protégées

- virtual void [\\_beforeValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator first-Token, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- virtual void [\\_afterValidation](#) ()

### Attributs protégés

- [QuantifiableContent](#) & [\\_content](#)
- [\\_StatesStack](#) [\\_stack](#)

## 6.19.1 Description détaillée

Définition à la ligne 19 du fichier QuantifiedContent.hh.

## 6.19.2 Documentation des définitions de type membres

- 6.19.2.1 `typedef std : :stack<_State> dtd : :QuantifiedContent : :_StatesStack [protected]`

Définition à la ligne 61 du fichier QuantifiedContent.hh.

## 6.19.3 Documentation des constructeurs et destructeur

- 6.19.3.1 `dtd : :QuantifiedContent : :QuantifiedContent ( QuantifiableContent & content )`

Définition à la ligne 53 du fichier QuantifiedContent.cpp.

- 6.19.3.2 `dtd : :QuantifiedContent : :~QuantifiedContent ( ) [virtual]`

Définition à la ligne 59 du fichier QuantifiedContent.cpp.



### 6.19.4 Documentation des fonctions membres

**6.19.4.1** `void dtd : :QuantifiedContent : :afterValidation ( ) [protected, virtual]`

Réimplémentée à partir de [dtd : :BrowsableContent](#).

Définition à la ligne 75 du fichier `QuantifiedContent.cpp`.

**6.19.4.2** `void dtd : :QuantifiedContent : :beforeValidation (`  
    `xml : :CompositeMarkupNode : :ChildrenIterator firstToken,`  
    `xml : :CompositeMarkupNode : :ChildrenIterator endToken, BrowsableContent *`  
    `nextStep ) [protected, virtual]`

Réimplémentée à partir de [dtd : :BrowsableContent](#).

Définition à la ligne 67 du fichier `QuantifiedContent.cpp`.

**6.19.4.3** `QuantifiableContent & dtd : :QuantifiedContent : :content ( )`

Définition à la ligne 39 du fichier `QuantifiedContent.cpp`.

**6.19.4.4** `const QuantifiableContent & dtd : :QuantifiedContent : :content ( ) const`

Définition à la ligne 44 du fichier `QuantifiedContent.cpp`.

**6.19.4.5** `virtual bool dtd : :QuantifiedContent : :validate ( const xml : :CompositeMarkupNode &`  
    `node ) [virtual]`

Réimplémentée à partir de [dtd : :BrowsableContent](#).

### 6.19.5 Documentation des données membres

**6.19.5.1** `QuantifiableContent& dtd : :QuantifiedContent : :_content [protected]`

Définition à la ligne 59 du fichier `QuantifiedContent.hh`.

**6.19.5.2** `_StatesStack dtd : :QuantifiedContent : :_stack [protected]`

Définition à la ligne 62 du fichier `QuantifiedContent.hh`.

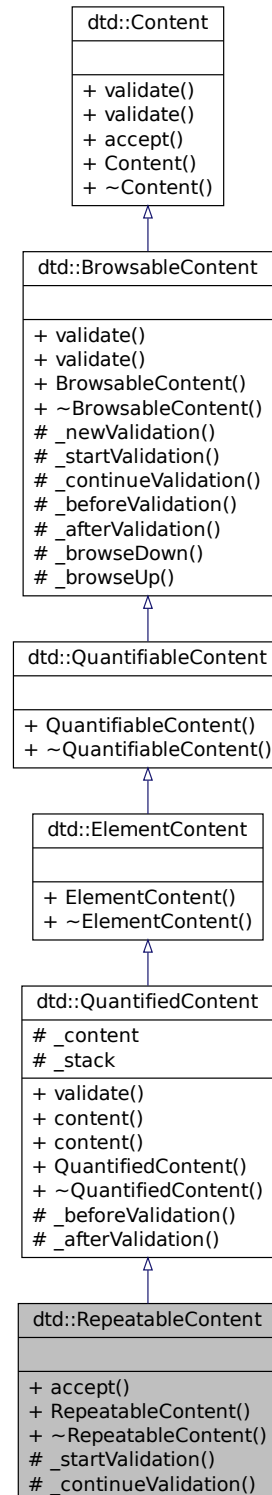
La documentation de cette classe a été générée à partir des fichiers suivants :

- [src/QuantifiedContent.hh](#)
- [src/QuantifiedContent.cpp](#)

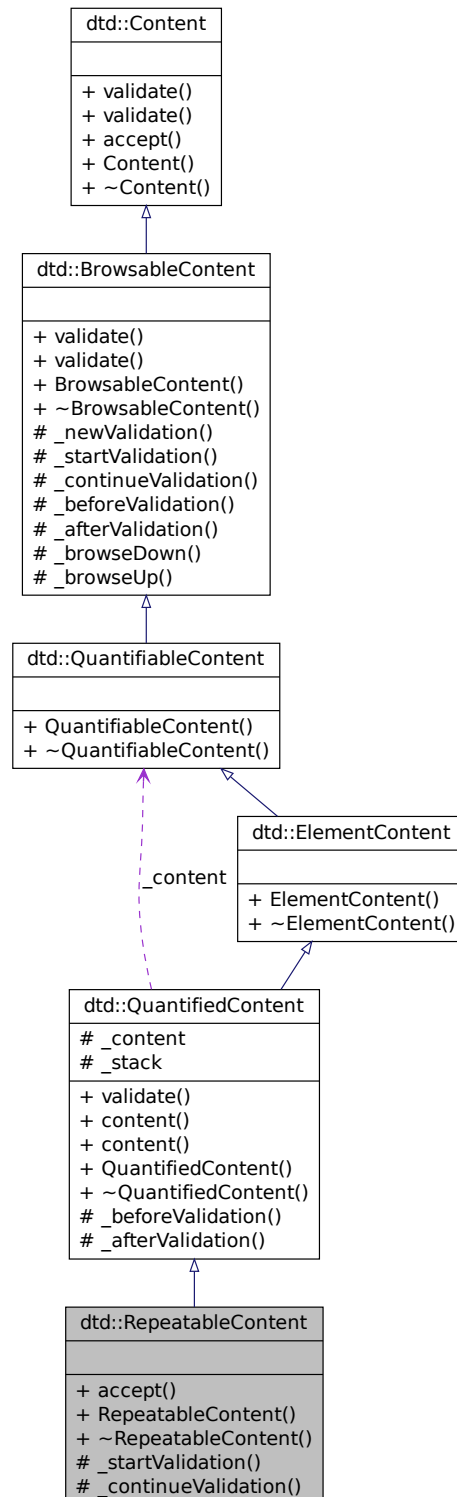
## 6.20 Référence de la classe dtd : :RepeatableContent

```
#include <RepeatableContent.hh>
```

Graphe d'héritage de dtd : :RepeatablContent :



Graphe de collaboration de dtd : :RepeatableContent :



### Fonctions membres publiques

- virtual void [accept](#) ([InterfaceDTDVisitor](#) &visitor) const
- [RepeatableContent](#) ([QuantifiableContent](#) &content)
- virtual [~RepeatableContent](#) ()

### Fonctions membres protégées

- virtual bool [\\_startValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- virtual bool [\\_continueValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator currentToken)

#### 6.20.1 Description détaillée

Définition à la ligne 18 du fichier RepeatableContent.hh.

#### 6.20.2 Documentation des constructeurs et destructeur

##### 6.20.2.1 dtd : :RepeatableContent : :RepeatableContent ( [QuantifiableContent](#) & *content* )

Définition à la ligne 42 du fichier RepeatableContent.cpp.

##### 6.20.2.2 dtd : :RepeatableContent : :~RepeatableContent ( ) [virtual]

Définition à la ligne 48 du fichier RepeatableContent.cpp.

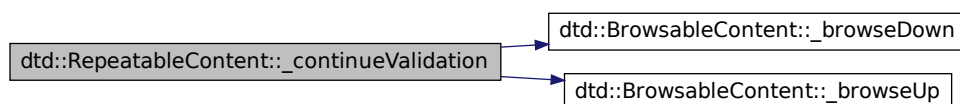
#### 6.20.3 Documentation des fonctions membres

##### 6.20.3.1 bool dtd : :RepeatableContent : :\_continueValidation ( xml : :CompositeMarkupNode : :ChildrenIterator *currentToken* ) [protected, virtual]

Réimplémentée à partir de [dtd : :BrowsableContent](#).

Définition à la ligne 67 du fichier RepeatableContent.cpp.

Voici le graphe d'appel pour cette fonction :



```
6.20.3.2 virtual bool dtd : :RepeatableContent : :startValidation (
    xml : :CompositeMarkupNode : :ChildrenIterator firstToken,
    xml : :CompositeMarkupNode : :ChildrenIterator endToken, BrowsableContent *
    nextStep ) [protected, virtual]
```

Implémente [dtd : :BrowsableContent](#).

```
6.20.3.3 void dtd : :RepeatableContent : :accept ( InterfaceDTDVisitor & visitor ) const
    [virtual]
```

Implémente [dtd : :Content](#).

Définition à la ligne 33 du fichier RepeatableContent.cpp.

Voici le graphe d'appel pour cette fonction :



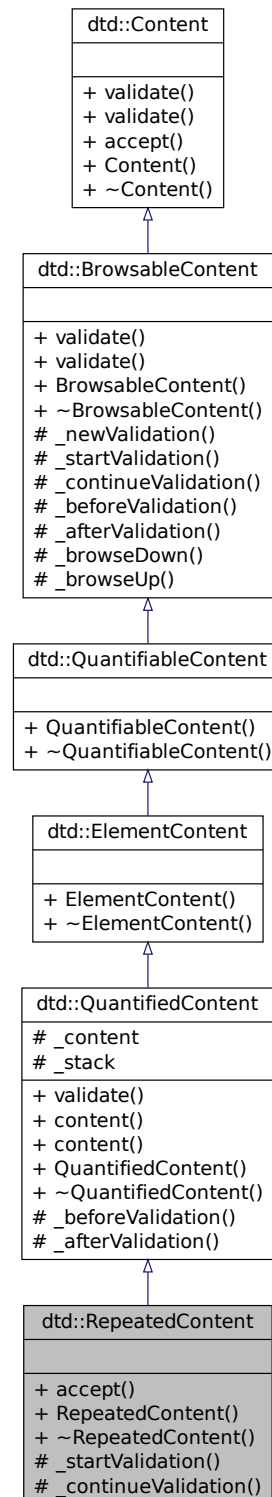
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/[RepeatableContent.hh](#)
- src/[RepeatableContent.cpp](#)

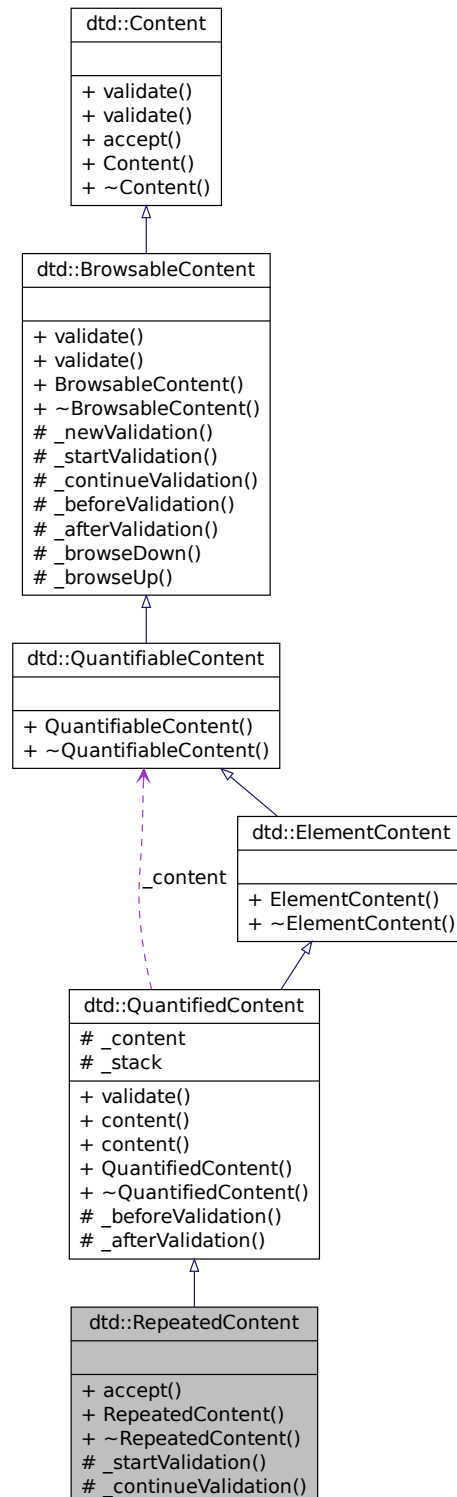
## 6.21 Référence de la classe dtd : :RepeatedContent

```
#include <RepeatedContent.hh>
```

Graphe d'héritage de dtd : :RepeatedContent :



Graphe de collaboration de dtd : :RepeatedContent :





## Fonctions membres publiques

- virtual void [accept](#) ([InterfaceDTDVisitor](#) &visitor) const
- [RepeatedContent](#) ([QuantifiableContent](#) &content)
- virtual [~RepeatedContent](#) ()

## Fonctions membres protégées

- virtual bool [\\_startValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- virtual bool [\\_continueValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator currentToken)

### 6.21.1 Description détaillée

Définition à la ligne 18 du fichier RepeatedContent.hh.

### 6.21.2 Documentation des constructeurs et destructeur

#### 6.21.2.1 dtd : :RepeatedContent : :RepeatedContent ( [QuantifiableContent](#) & *content* )

Définition à la ligne 42 du fichier RepeatedContent.cpp.

#### 6.21.2.2 dtd : :RepeatedContent : :~RepeatedContent ( ) [virtual]

Définition à la ligne 48 du fichier RepeatedContent.cpp.

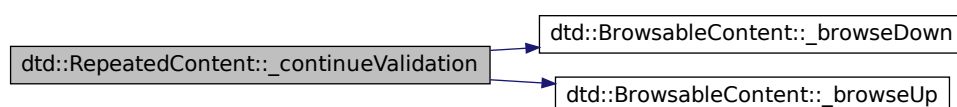
### 6.21.3 Documentation des fonctions membres

#### 6.21.3.1 bool dtd : :RepeatedContent : :\_continueValidation ( xml : :CompositeMarkupNode : :ChildrenIterator *currentToken* ) [protected, virtual]

Réimplémentée à partir de [dtd : :BrowsableContent](#).

Définition à la ligne 65 du fichier RepeatedContent.cpp.

Voici le graphe d'appel pour cette fonction :



```
6.21.3.2 virtual bool dtd : :RepeatedContent : :startValidation (
    xml : :CompositeMarkupNode : :ChildrenIterator firstToken,
    xml : :CompositeMarkupNode : :ChildrenIterator endToken, BrowsableContent *
    nextStep ) [protected, virtual]
```

Implémente [dtd : :BrowsableContent](#).

```
6.21.3.3 void dtd : :RepeatedContent : :accept ( InterfaceDTDVisitor & visitor ) const
    [virtual]
```

Implémente [dtd : :Content](#).

Définition à la ligne 33 du fichier RepeatedContent.cpp.

Voici le graphe d'appel pour cette fonction :



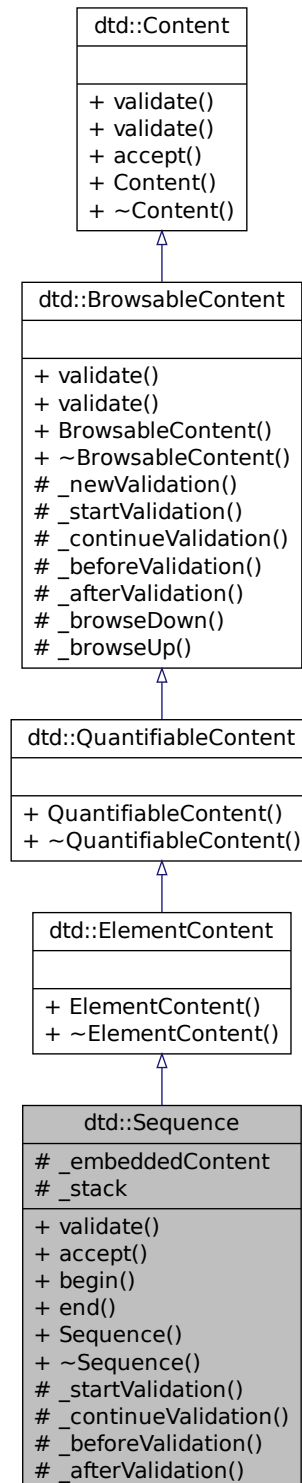
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/[RepeatedContent.hh](#)
- src/[RepeatedContent.cpp](#)

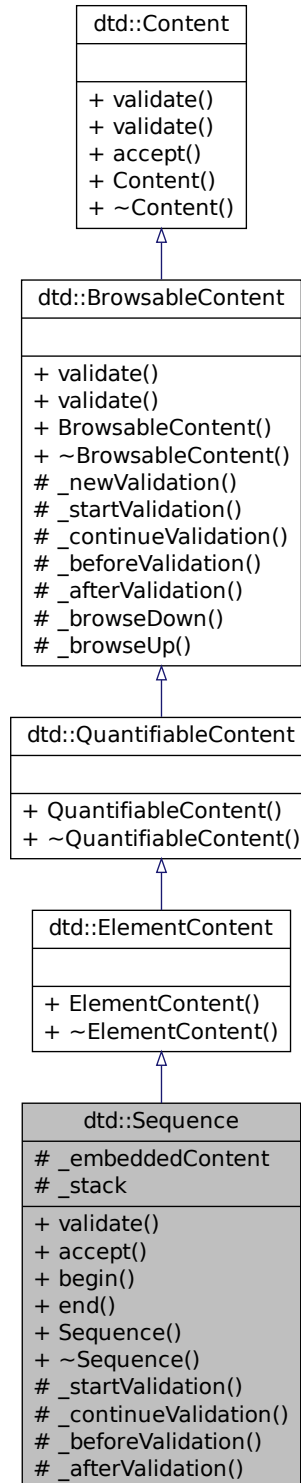
## 6.22 Référence de la classe dtd : :Sequence

```
#include <Sequence.hh>
```

Graphe d'héritage de dtd : :Sequence :



Graphe de collaboration de dtd : :Sequence :



## Classes

- struct [\\_State](#)

## Types publics

- typedef std : :list< [ElementContent](#) \* > [OrderedContent](#)
- typedef [\\_OrderedContent](#) : :const\_iterator [const\\_iterator](#)

## Fonctions membres publiques

- virtual bool [validate](#) (const xml : :CompositeMarkupNode &node)
- virtual void [accept](#) ([InterfaceDTDVisitor](#) &visitor) const
- [const\\_iterator begin](#) () const
- [const\\_iterator end](#) () const
- [Sequence](#) (const [OrderedContent](#) &embeddedContent)
- virtual [~Sequence](#) ()

## Types protégés

- typedef std : :list< [ElementContent](#) \* > [\\_OrderedContent](#)
- typedef std : :stack< [\\_State](#) > [\\_StatesStack](#)

## Fonctions membres protégées

- virtual bool [\\_startValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- virtual bool [\\_continueValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator currentToken)
- virtual void [\\_beforeValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- virtual void [\\_afterValidation](#) ()

## Attributs protégés

- [\\_OrderedContent](#) [\\_embeddedContent](#)
- [\\_StatesStack](#) [\\_stack](#)

### 6.22.1 Description détaillée

Définition à la ligne 20 du fichier Sequence.hh.

## 6.22.2 Documentation des définitions de type membres

**6.22.2.1** `typedef std : :list<ElementContent*> dtd : :Sequence : :_OrderedContent  
[protected]`

Définition à la ligne 23 du fichier Sequence.hh.

**6.22.2.2** `typedef std : :stack<_State> dtd : :Sequence : :_StatesStack [protected]`

Définition à la ligne 97 du fichier Sequence.hh.

**6.22.2.3** `typedef _OrderedContent : :const_iterator dtd : :Sequence : :const_iterator`

Définition à la ligne 29 du fichier Sequence.hh.

**6.22.2.4** `typedef std : :list<ElementContent*> dtd : :Sequence : :OrderedContent`

Définition à la ligne 28 du fichier Sequence.hh.

## 6.22.3 Documentation des constructeurs et destructeur

**6.22.3.1** `dtd : :Sequence : :Sequence ( const OrderedContent & embeddedContent )`

Définition à la ligne 61 du fichier Sequence.cpp.

**6.22.3.2** `dtd : :Sequence : :~Sequence ( ) [virtual]`

Définition à la ligne 67 du fichier Sequence.cpp.

## 6.22.4 Documentation des fonctions membres

**6.22.4.1** `void dtd : :Sequence : :afterValidation ( ) [protected, virtual]`

Réimplémentée à partir de [dtd : :BrowsableContent](#).

Définition à la ligne 88 du fichier Sequence.cpp.

**6.22.4.2** `void dtd : :Sequence : :beforeValidation ( xml : :CompositeMarkupNode : :ChildrenIt-  
erator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken,  
BrowsableContent * nextStep ) [protected, virtual]`

Réimplémentée à partir de [dtd : :BrowsableContent](#).

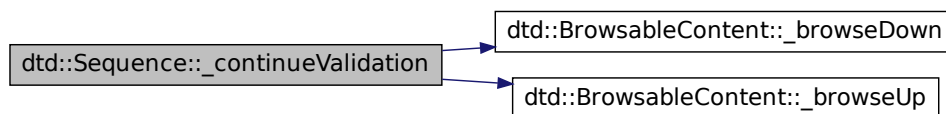
Définition à la ligne 79 du fichier Sequence.cpp.

**6.22.4.3** `bool dtd : :Sequence : :continueValidation ( xml : :CompositeMarkupNode : :ChildrenIterator currentToken )` `[protected, virtual]`

Réimplémentée à partir de [dtd : :BrowsableContent](#).

Définition à la ligne 102 du fichier Sequence.cpp.

Voici le graphe d'appel pour cette fonction :



**6.22.4.4** `virtual bool dtd : :Sequence : :startValidation ( xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, BrowsableContent * nextStep )` `[protected, virtual]`

Implémente [dtd : :BrowsableContent](#).

**6.22.4.5** `void dtd : :Sequence : :accept ( InterfaceDTDVisitor & visitor ) const` `[virtual]`

Implémente [dtd : :Content](#).

Définition à la ligne 41 du fichier Sequence.cpp.

Voici le graphe d'appel pour cette fonction :



**6.22.4.6** `Sequence : :const_iterator dtd : :Sequence : :begin ( ) const`

Définition à la ligne 46 du fichier Sequence.cpp.

#### 6.22.4.7 `Sequence : :const_iterator dtd : :Sequence : :end ( ) const`

Définition à la ligne 51 du fichier `Sequence.cpp`.

#### 6.22.4.8 `virtual bool dtd : :Sequence : :validate ( const xml : :CompositeMarkupNode & node ) [virtual]`

Réimplémentée à partir de [dtd : :BrowsableContent](#).

### 6.22.5 Documentation des données membres

#### 6.22.5.1 `_OrderedContent dtd : :Sequence : :_embeddedContent [protected]`

Définition à la ligne 81 du fichier `Sequence.hh`.

#### 6.22.5.2 `_StatesStack dtd : :Sequence : :_stack [protected]`

Définition à la ligne 98 du fichier `Sequence.hh`.

La documentation de cette classe a été générée à partir des fichiers suivants :

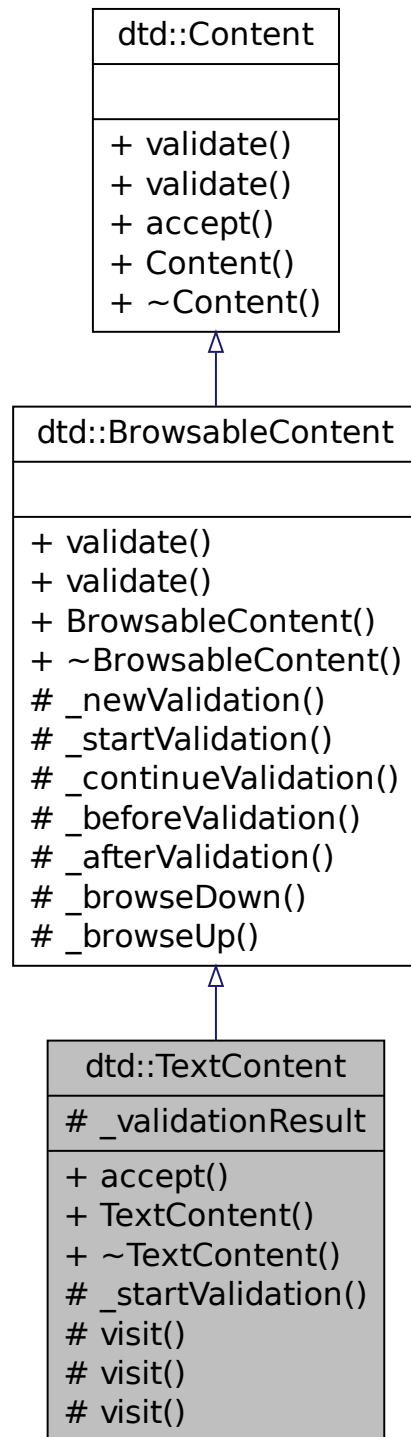
- [src/Sequence.hh](#)
- [src/Sequence.cpp](#)

## 6.23 Référence de la classe `dtd : :TextContent`

```
#include <TextContent.hh>
```



Graphe d'héritage de dtd : :TextContent :



Graphe de collaboration de dtd : :TextContent :



### Fonctions membres publiques

- virtual void [accept](#) ([InterfaceDTDVisitor](#) &visitor) const
- [TextContent](#) ()
- virtual [~TextContent](#) ()

### Fonctions membres protégées

- virtual bool [\\_startValidation](#) (xml : :CompositeMarkupNode : :ChildrenIterator firstToken, xml : :CompositeMarkupNode : :ChildrenIterator endToken, [BrowsableContent](#) \*nextStep)
- virtual void [visit](#) (const xml : :TextNode &node)
- virtual void [visit](#) (const xml : :MarkupNode &node)
- virtual void [visit](#) (const xml : :CompositeMarkupNode &node)

### Attributs protégés

- bool [\\_validationResult](#)

#### 6.23.1 Description détaillée

Définition à la ligne 19 du fichier TextContent.hh.

#### 6.23.2 Documentation des constructeurs et destructeur

##### 6.23.2.1 dtd : :TextContent : :TextContent ( )

Définition à la ligne 42 du fichier TextContent.cpp.

##### 6.23.2.2 dtd : :TextContent : :~TextContent ( ) [virtual]

Définition à la ligne 47 du fichier TextContent.cpp.

#### 6.23.3 Documentation des fonctions membres

##### 6.23.3.1 virtual bool dtd : :TextContent : :\_startValidation ( xml : :CompositeMarkupNode : :ChildrenIterator *firstToken*, xml : :CompositeMarkupNode : :ChildrenIterator *endToken*, [BrowsableContent](#) \**nextStep* ) [protected, virtual]

Implémente [dtd : :BrowsableContent](#).

##### 6.23.3.2 void dtd : :TextContent : :accept ( [InterfaceDTDVisitor](#) & *visitor* ) const [virtual]

Implémente [dtd : :Content](#).

Définition à la ligne 33 du fichier `TextContent.cpp`.

Voici le graphe d'appel pour cette fonction :



**6.23.3.3** `virtual void dtd : :TextContent : :visit ( const xml : :MarkupNode & node )`  
[protected, virtual]

**6.23.3.4** `virtual void dtd : :TextContent : :visit ( const xml : :TextNode & node )`  
[protected, virtual]

**6.23.3.5** `virtual void dtd : :TextContent : :visit ( const xml : :CompositeMarkupNode & node )`  
[protected, virtual]

## 6.23.4 Documentation des données membres

**6.23.4.1** `bool dtd : :TextContent : :_validationResult` [protected]

Définition à la ligne 53 du fichier `TextContent.hh`.

La documentation de cette classe a été générée à partir des fichiers suivants :

- src/[TextContent.hh](#)
- src/[TextContent.cpp](#)

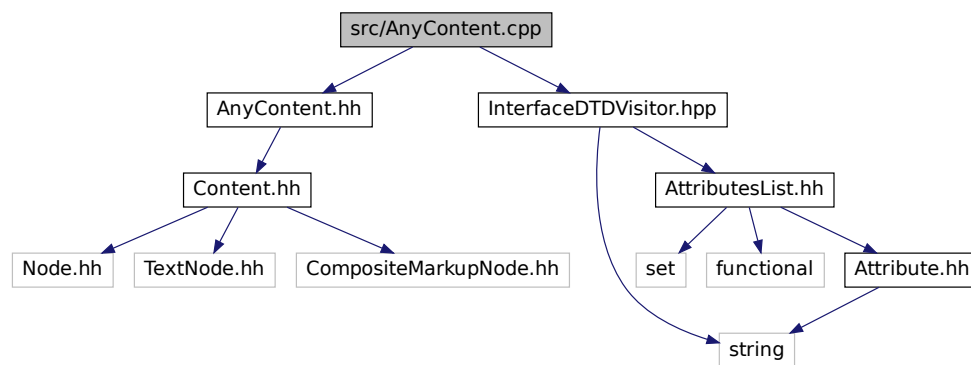
## Chapitre 7

# Documentation des fichiers

### 7.1 Référence du fichier src/AnyContent.cpp

```
#include "AnyContent.hh"  
#include "InterfaceDTDVisitor.hpp"
```

Graphe des dépendances par inclusion de AnyContent.cpp :



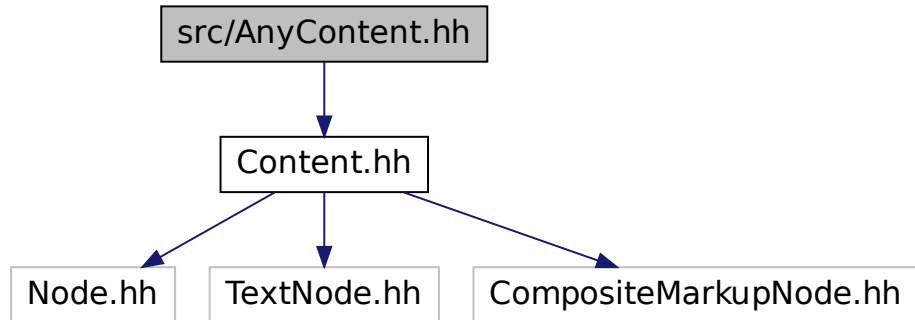
### Espaces de nommage

– namespace [dtd](#)

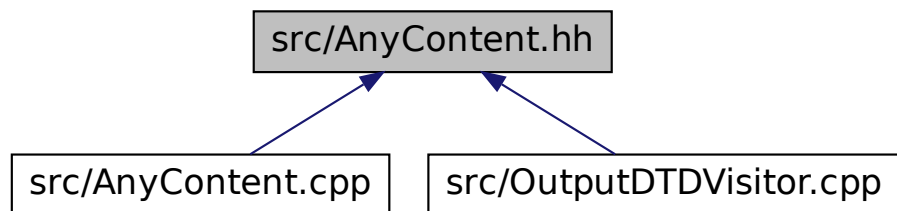
### 7.2 Référence du fichier src/AnyContent.hh

```
#include "Content.hh"
```

Graphe des dépendances par inclusion de AnyContent.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

– class `dtd : AnyContent`

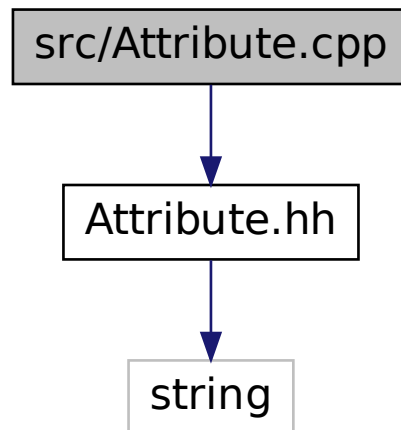
### Espaces de nommage

– namespace `dtd`

## 7.3 Référence du fichier src/Attribute.cpp

```
#include "Attribute.hh"
```

Graphe des dépendances par inclusion de Attribute.cpp :



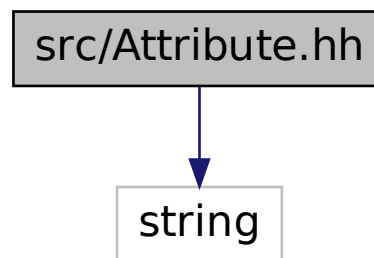
### Espaces de nommage

– namespace [dtd](#)

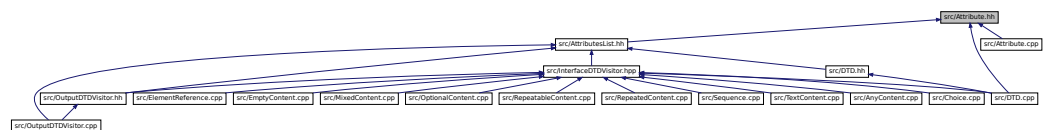
## 7.4 Référence du fichier src/Attribute.hh

```
#include <string>
```

Graphe des dépendances par inclusion de Attribute.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

– class `dtd :Attribute`

## Espaces de nommage

– namespace `dtd`

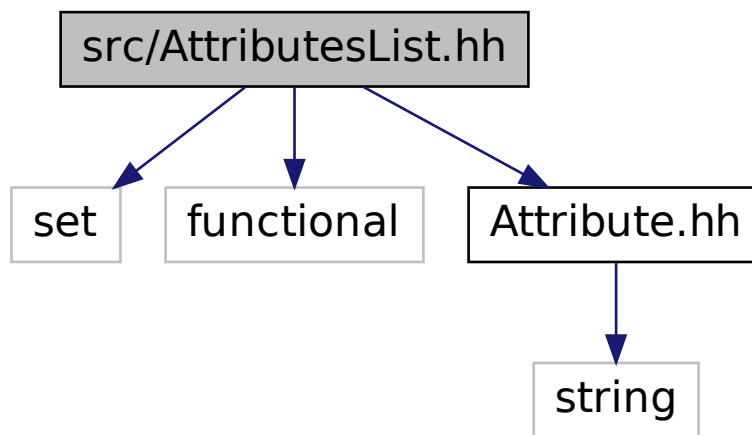
## 7.5 Référence du fichier `src/AttributesList.hh`

```

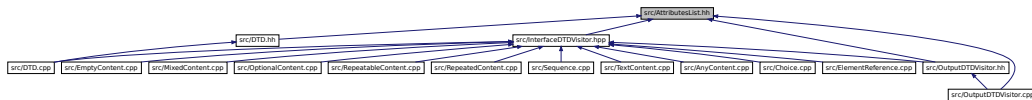
#include <set>
#include <functional>
#include "Attribute.hh"
  
```



Grphe des dépendances par inclusion de AttributesList.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

- struct `dtd : :AttributesComparator`

## Espaces de nommage

- namespace `dtd`

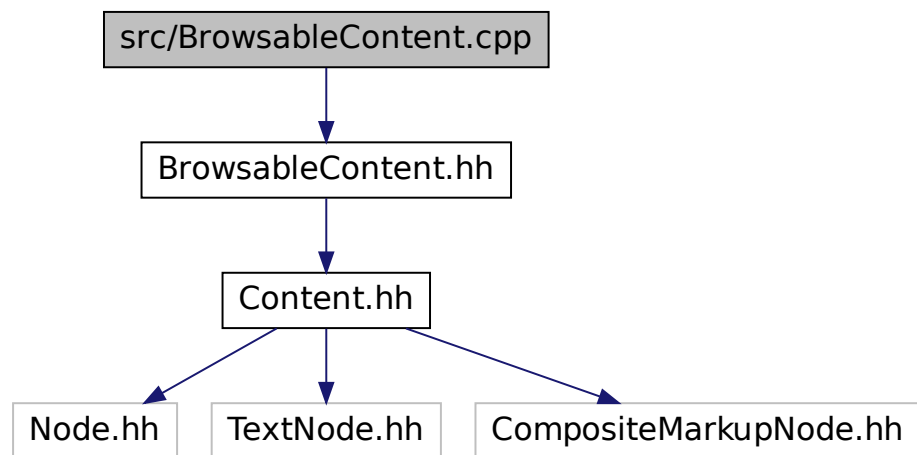
## Définition de type

- typedef `std : :set< Attribute *, AttributesComparator > dtd : :AttributesList`

## 7.6 Référence du fichier src/BrowsableContent.cpp

```
#include "BrowsableContent.hh"
```

Graph des dépendances par inclusion de BrowsableContent.cpp :



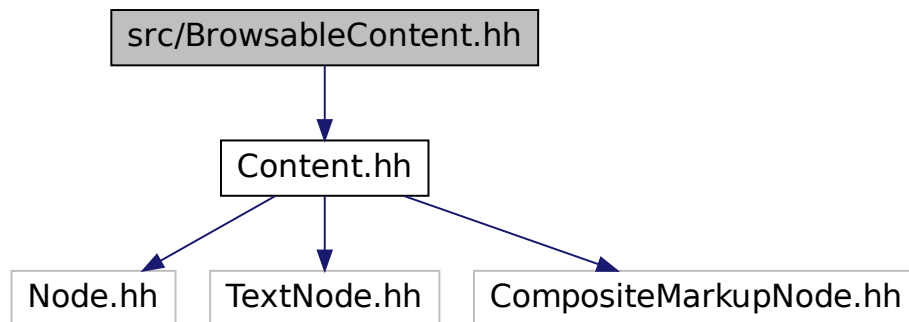
### Espaces de nommage

– namespace [dtd](#)

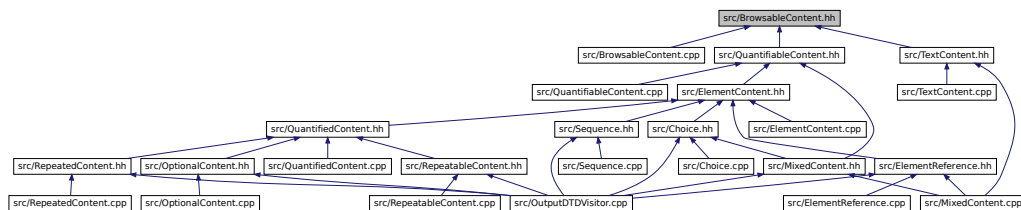
## 7.7 Référence du fichier src/BrowsableContent.hh

```
#include "Content.hh"
```

Grphe des dépendances par inclusion de BrowsableContent.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

– class `dtd` : `BrowsableContent`

## Espaces de nommage

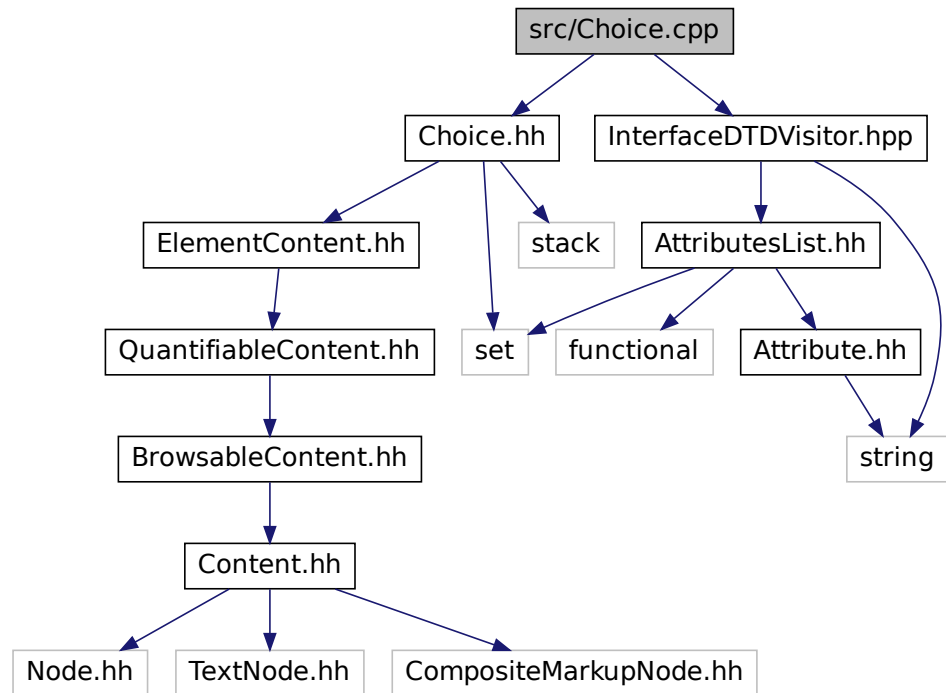
– namespace `dtd`

## 7.8 Référence du fichier src/Choice.cpp

```

#include "Choice.hh"
#include "InterfaceDTDVisitor.hpp"
  
```

Graphe des dépendances par inclusion de Choice.cpp :



## Espaces de nommage

– namespace `dtd`

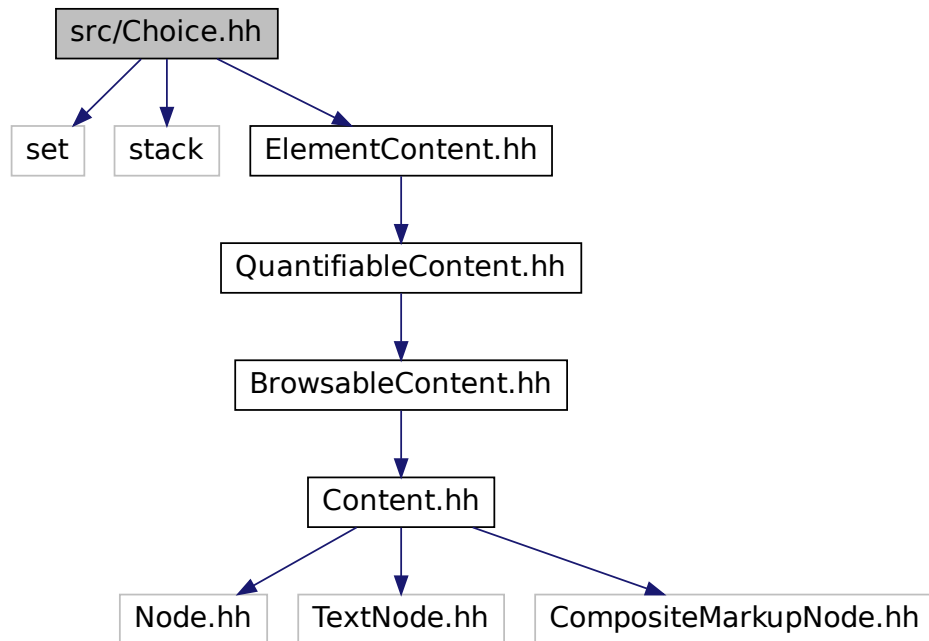
## 7.9 Référence du fichier `src/Choice.hh`

```

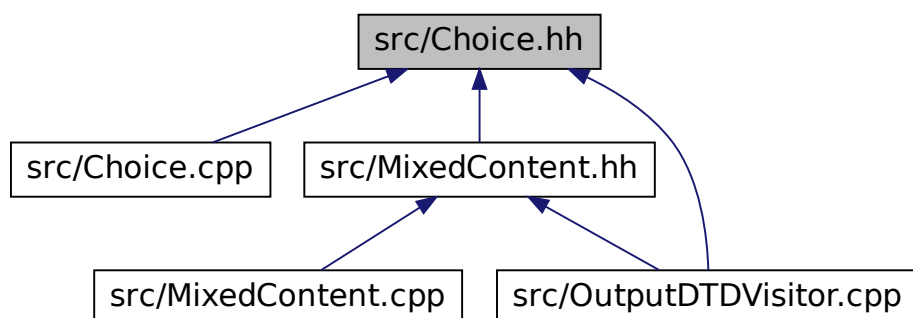
#include <set>
#include <stack>
#include "ElementContent.hh"

```

Graphe des dépendances par inclusion de Choice.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

- class `dtd : :Choice`
- struct `dtd : :Choice : :_State`

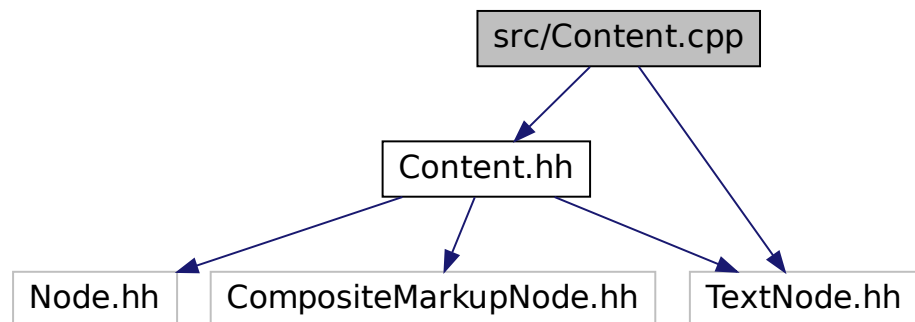
### Espaces de nommage

- namespace `dtd`

## 7.10 Référence du fichier `src/Content.cpp`

```
#include "Content.hh"  
#include "TextNode.hh"
```

Graphes des dépendances par inclusion de `Content.cpp` :



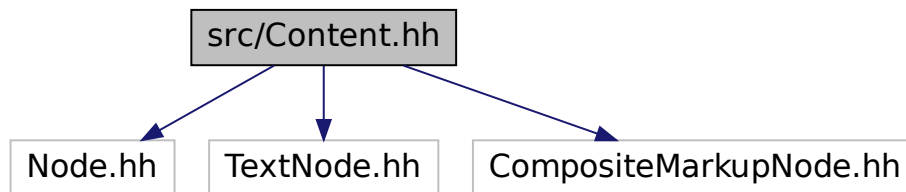
### Espaces de nommage

- namespace `dtd`

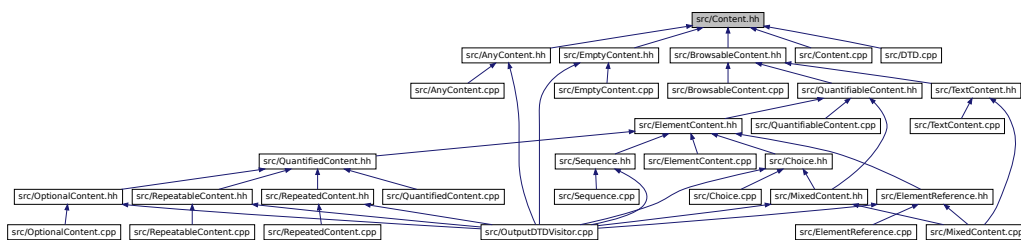
## 7.11 Référence du fichier `src/Content.hh`

```
#include "Node.hh"  
#include "TextNode.hh"  
#include "CompositeMarkupNode.hh"
```

Grphe des dépendances par inclusion de Content.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

– class `dtd : :Content`

## Espaces de nommage

– namespace `dtd`

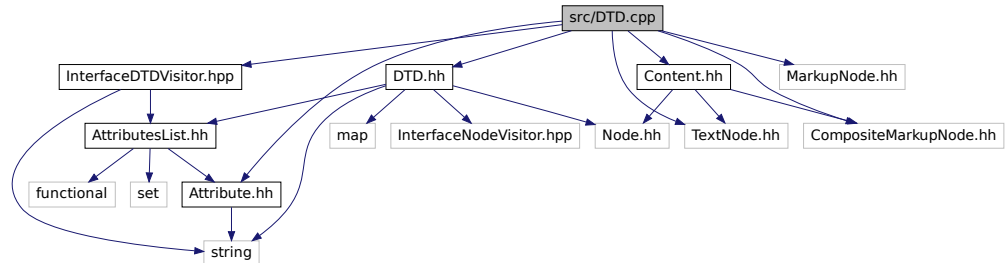
## 7.12 Référence du fichier src/DTD.cpp

```

#include "DTD.hh"
#include "Content.hh"
#include "Attribute.hh"
#include "MarkupNode.hh"
#include "TextNode.hh"
#include "CompositeMarkupNode.hh"
  
```

```
#include "InterfaceDTDVisitor.hpp"
```

Graphes des dépendances par inclusion de DTD.cpp :



## Espaces de nommage

– namespace `dtd`

## 7.13 Référence du fichier src/DTD.hh

```
#include <string>
```

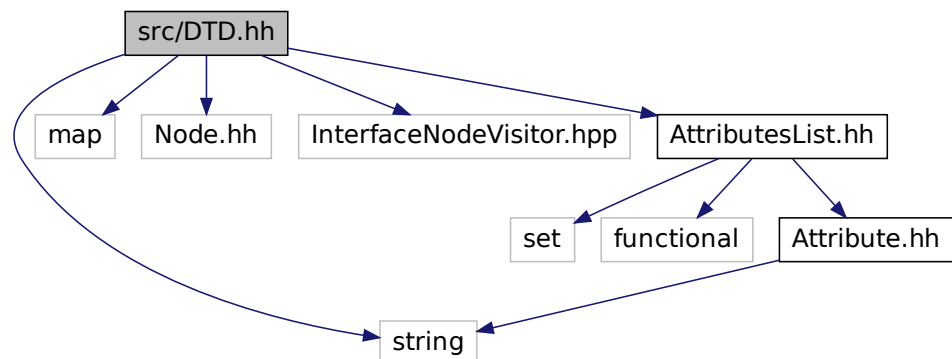
```
#include <map>
```

```
#include "Node.hh"
```

```
#include "InterfaceNodeVisitor.hpp"
```

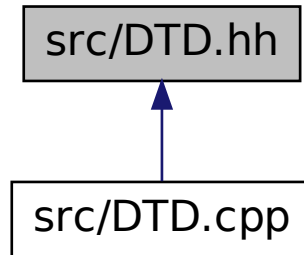
```
#include "AttributesList.hh"
```

Graphes des dépendances par inclusion de DTD.hh :





Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

– class `dtd` : `:DTD`

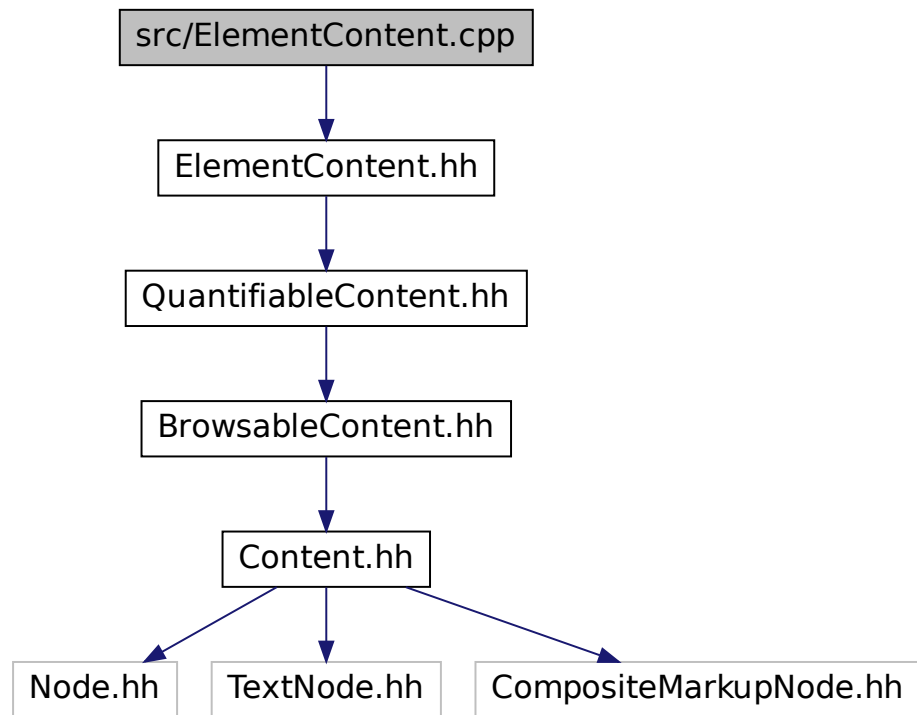
### Espaces de nommage

– namespace `dtd`

## 7.14 Référence du fichier src/ElementContent.cpp

```
#include "ElementContent.hh"
```

Graphe des dépendances par inclusion de ElementContent.cpp :



### Espaces de nommage

– namespace [dtd](#)

## 7.15 Référence du fichier src/ElementContent.hh

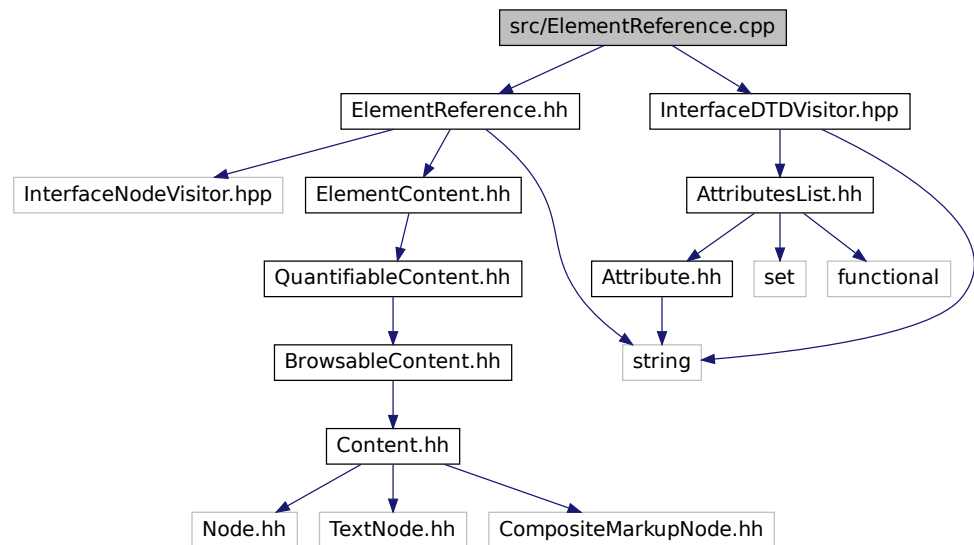
```
#include "QuantifiableContent.hh"
```



## 7.16 Référence du fichier src/ElementReference.cpp

```
#include "ElementReference.hh"
#include "InterfaceDTDVisitor.hpp"
```

Graphes des dépendances par inclusion de ElementReference.cpp :



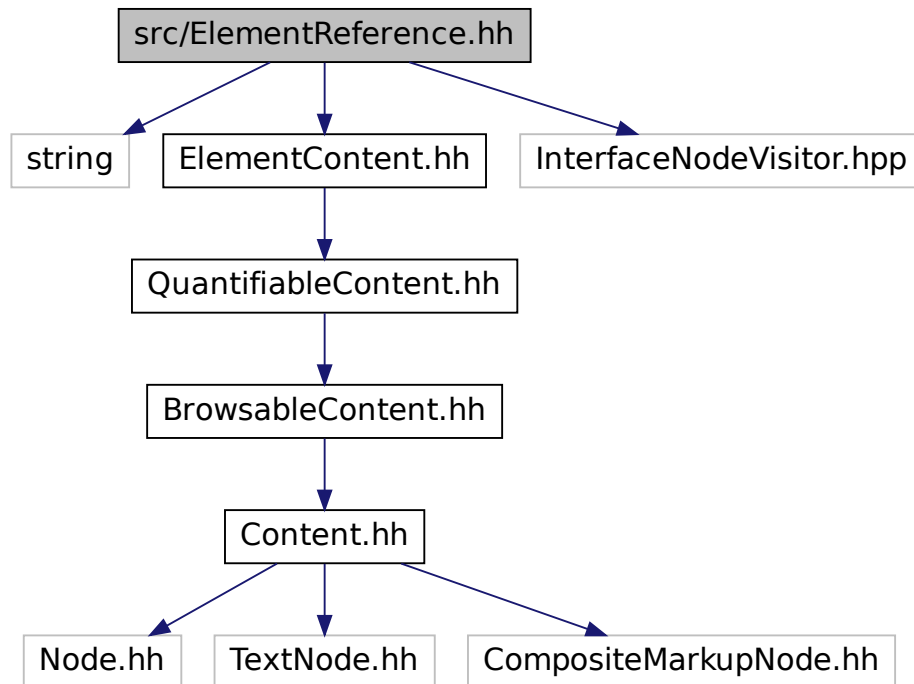
### Espaces de nommage

– namespace `dtd`

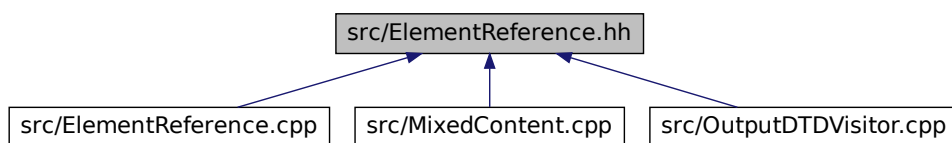
## 7.17 Référence du fichier src/ElementReference.hh

```
#include <string>
#include "ElementContent.hh"
#include "InterfaceNodeVisitor.hpp"
```

Graphe des dépendances par inclusion de ElementReference.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

– class `dtd` : [ElementReference](#)

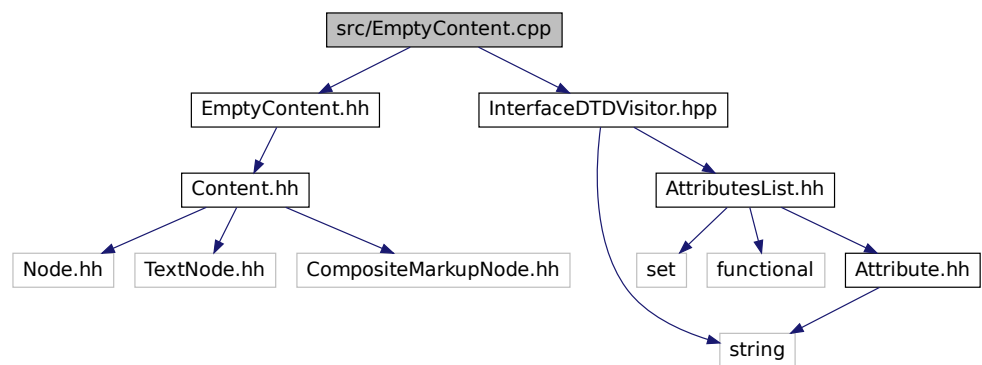
## Espaces de nommage

– namespace [dtd](#)

## 7.18 Référence du fichier src/EmptyContent.cpp

```
#include "EmptyContent.hh"  
#include "InterfaceDTDVisitor.hpp"
```

Graphes des dépendances par inclusion de EmptyContent.cpp :



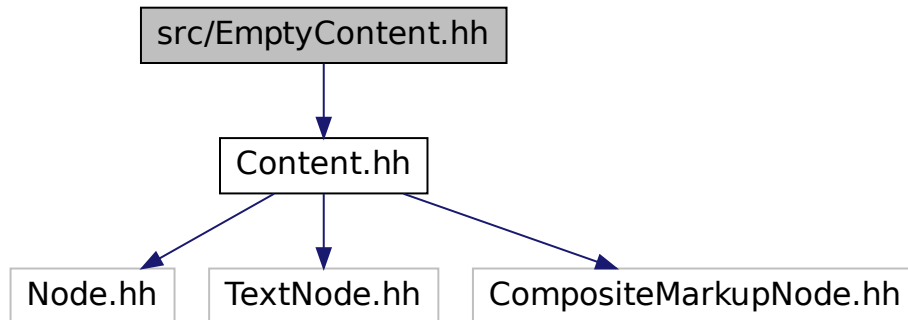
## Espaces de nommage

– namespace [dtd](#)

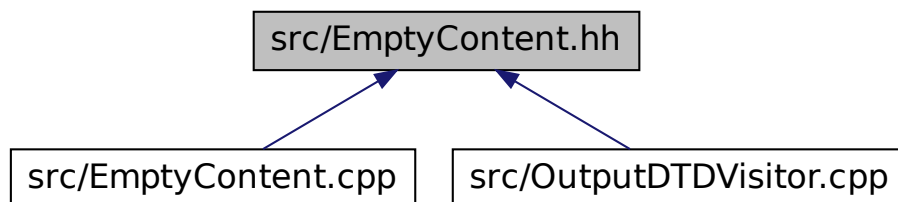
## 7.19 Référence du fichier src/EmptyContent.hh

```
#include "Content.hh"
```

Graphe des dépendances par inclusion de EmptyContent.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

– class `dtd : EmptyContent`

### Espaces de nommage

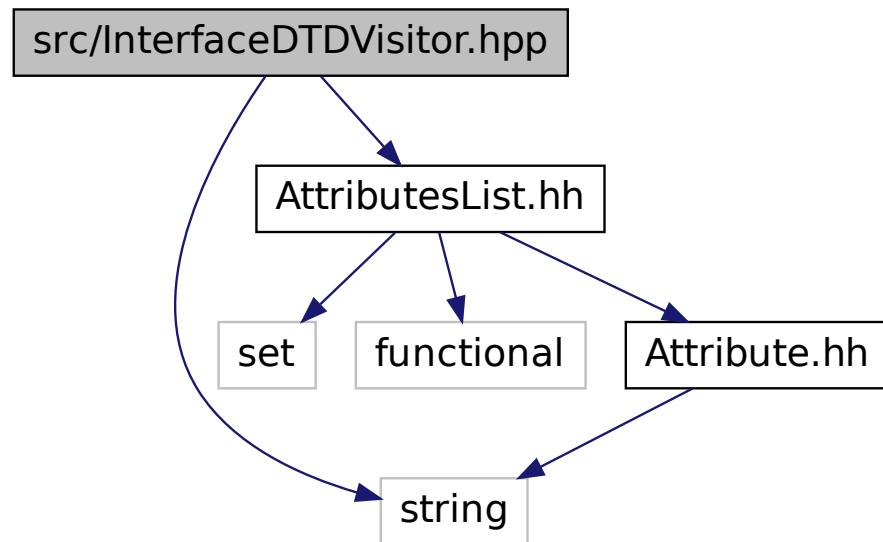
– namespace `dtd`

## 7.20 Référence du fichier src/InterfaceDTDVisitor.hpp

```
#include <string>
```

```
#include "AttributesList.hh"
```

Graphe des dépendances par inclusion de InterfaceDTDVisitor.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

– class `dtd` : `InterfaceDTDVisitor`

## Espaces de nommage

– namespace `dtd`

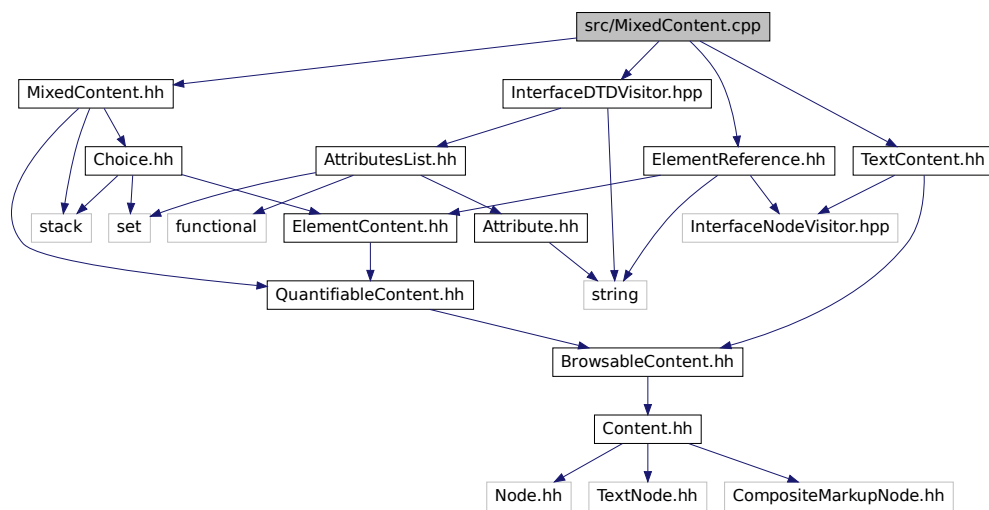
## 7.21 Référence du fichier src/MixedContent.cpp

```
#include "MixedContent.hh"
```



```
#include "TextContent.hh"
#include "ElementReference.hh"
#include "InterfaceDTDVisitor.hpp"
```

Graphe des dépendances par inclusion de MixedContent.cpp :



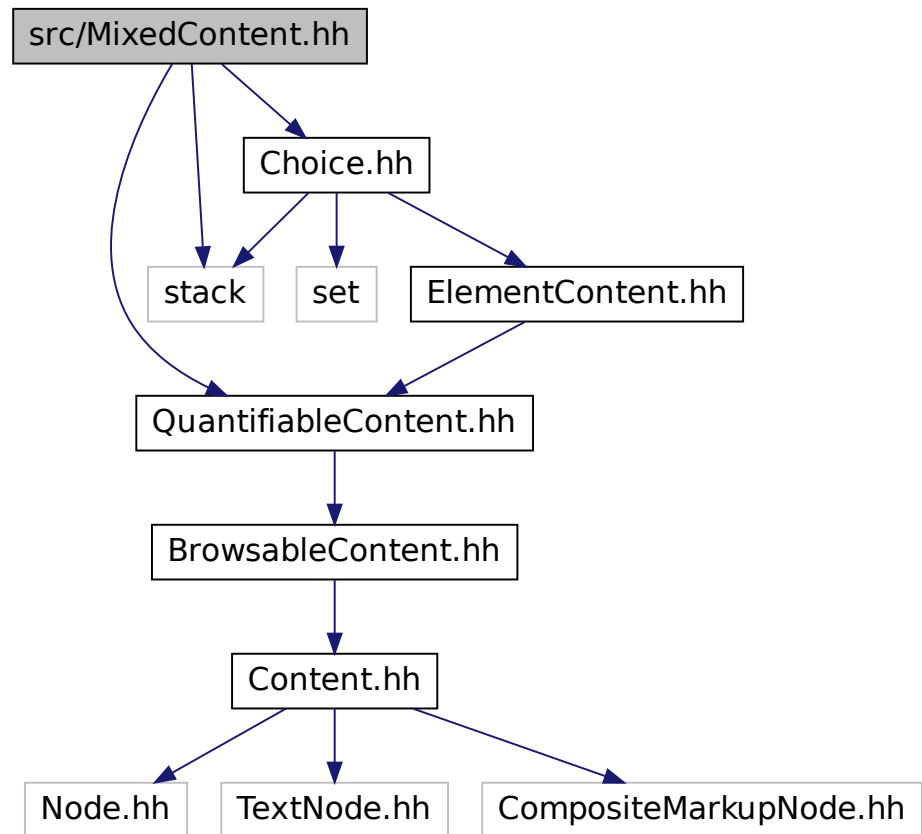
## Espaces de nommage

– namespace [dtd](#)

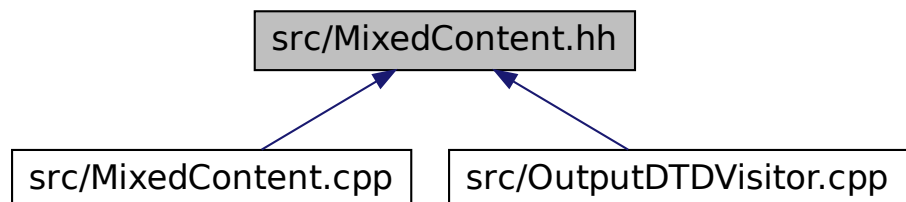
## 7.22 Référence du fichier src/MixedContent.hh

```
#include <stack>
#include "QuantifiableContent.hh"
#include "Choice.hh"
```

Graphe des dépendances par inclusion de MixedContent.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



**Classes**

- class `dtd` : `MixedContent`

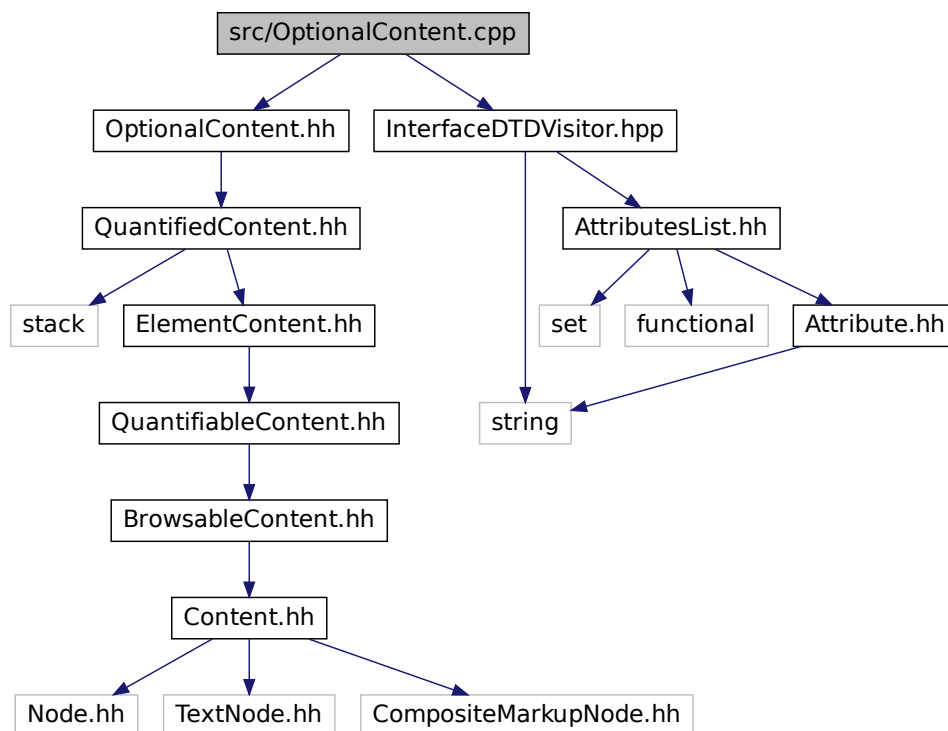
**Espaces de nommage**

- namespace `dtd`

**7.23 Référence du fichier src/OptionalContent.cpp**

```
#include "OptionalContent.hh"
#include "InterfaceDTDVisitor.hpp"
```

Graphes des dépendances par inclusion de OptionalContent.cpp :

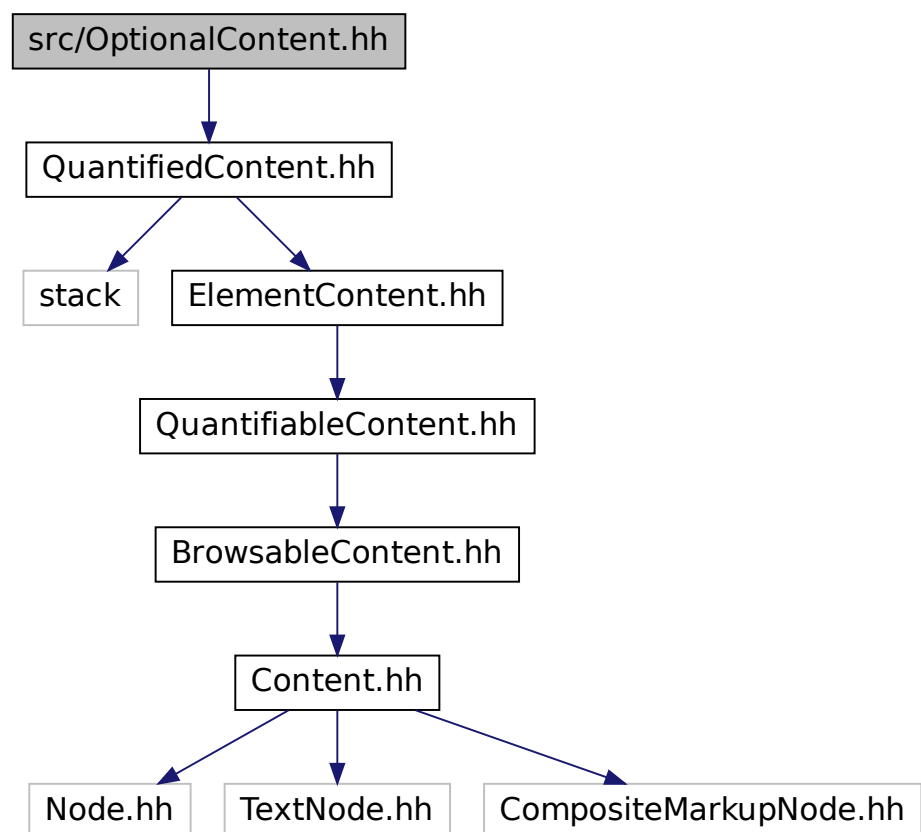
**Espaces de nommage**

- namespace `dtd`

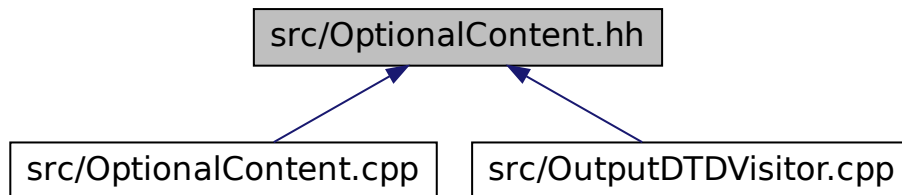
## 7.24 Référence du fichier src/OptionalContent.hh

```
#include "QuantifiedContent.hh"
```

Graphe des dépendances par inclusion de OptionalContent.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

– class `dtd : OptionalContent`

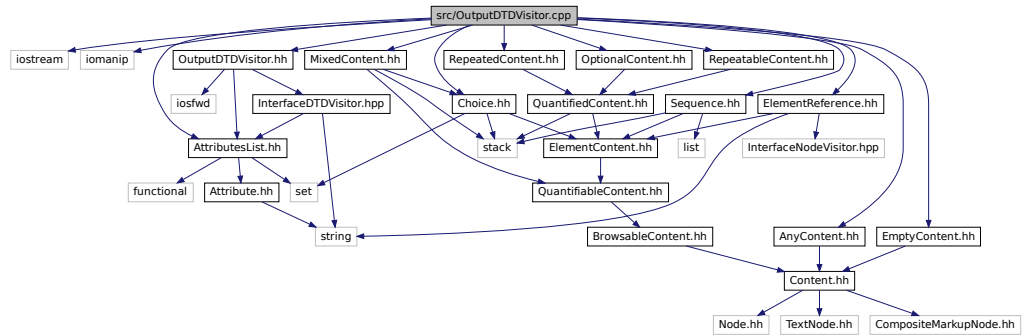
### Espaces de nommage

– namespace `dtd`

## 7.25 Référence du fichier src/OutputDTDVisitor.cpp

```
#include <iostream>
#include <iomanip>
#include "OutputDTDVisitor.hh"
#include "AttributesList.hh"
#include "EmptyContent.hh"
#include "AnyContent.hh"
#include "MixedContent.hh"
#include "ElementReference.hh"
#include "Choice.hh"
#include "Sequence.hh"
#include "OptionalContent.hh"
#include "RepeatableContent.hh"
#include "RepeatedContent.hh"
```

Grphe des dpendances par inclusion de OutputDTDVisitor.cpp :



## Espaces de nommage

– namespace `dtd`

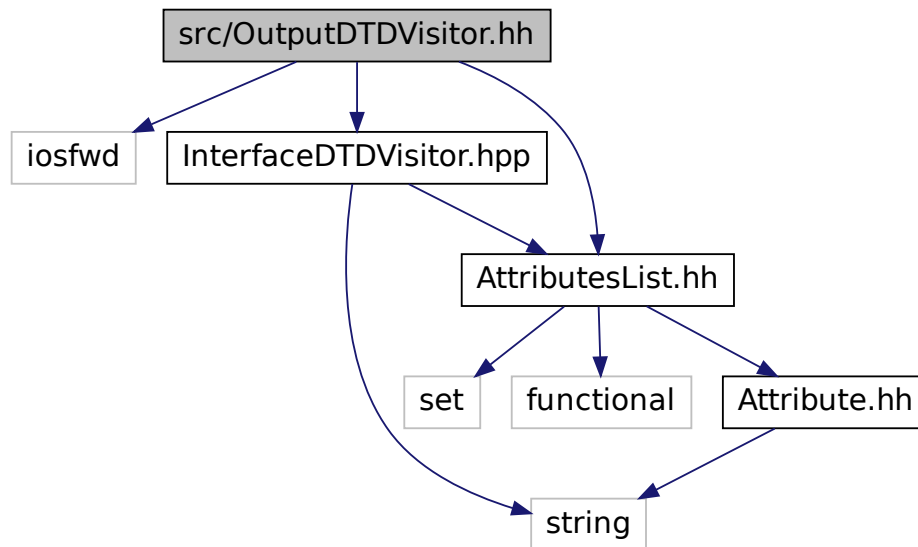
## 7.26 Référence du fichier src/OutputDTDVisitor.hh

```

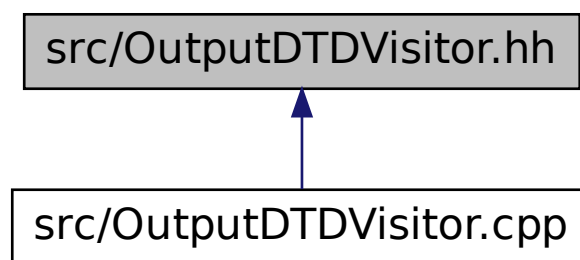
#include <iosfwd>
#include "InterfaceDTDVisitor.hpp"
#include "AttributesList.hh"

```

Graphe des dépendances par inclusion de OutputDTDVisitor.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

— class `dtd` : `OutputDTDVisitor`

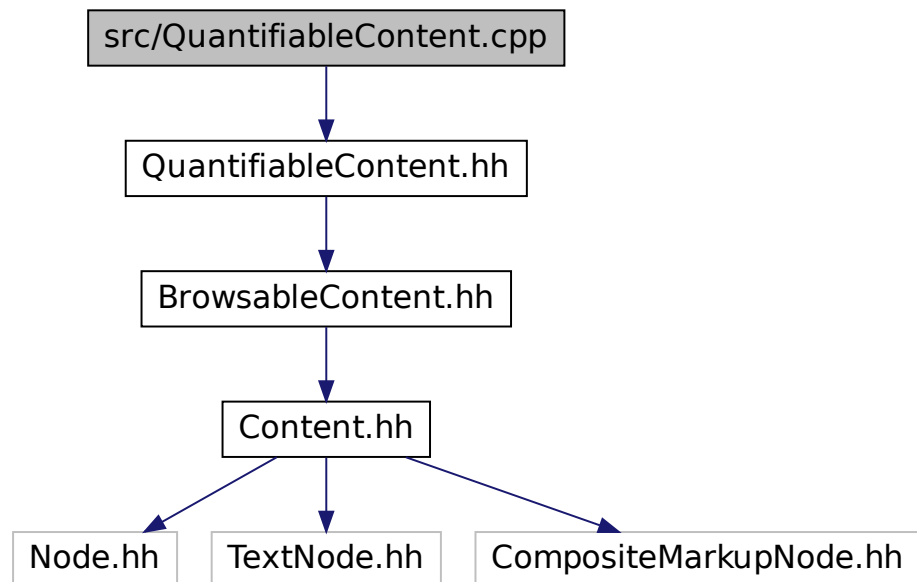
## Espaces de nommage

– namespace `dtd`

## 7.27 Référence du fichier `src/QuantifiableContent.cpp`

```
#include "QuantifiableContent.hh"
```

Graphes des dépendances par inclusion de `QuantifiableContent.cpp` :



## Espaces de nommage

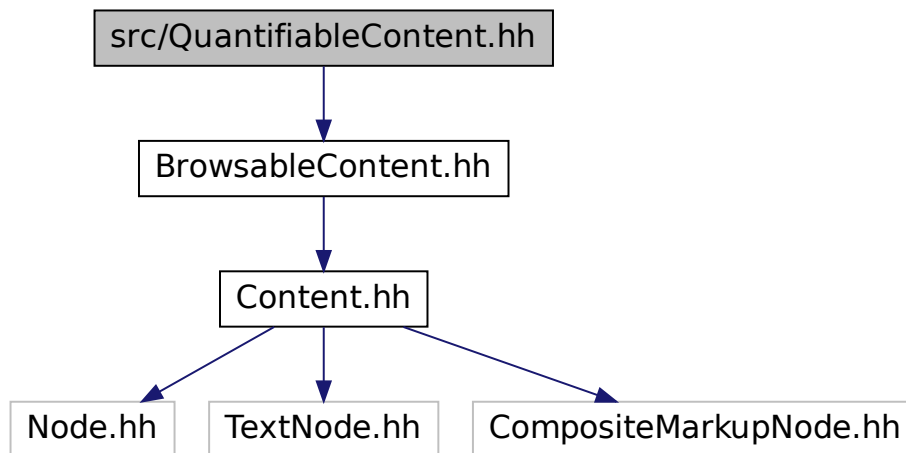
– namespace `dtd`

## 7.28 Référence du fichier `src/QuantifiableContent.hh`

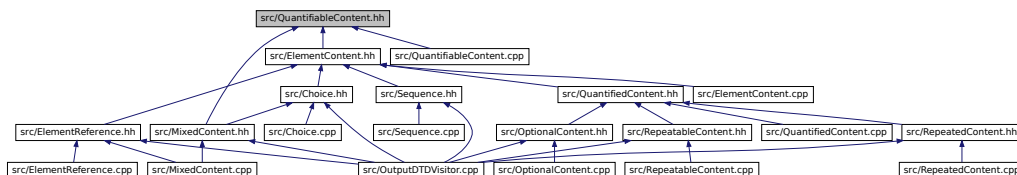
```
#include "BrowseableContent.hh"
```



Graphe des dépendances par inclusion de QuantifiableContent.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

- class `dtd : :QuantifiableContent`
- struct `dtd : :QuantifiableContent : :_State`

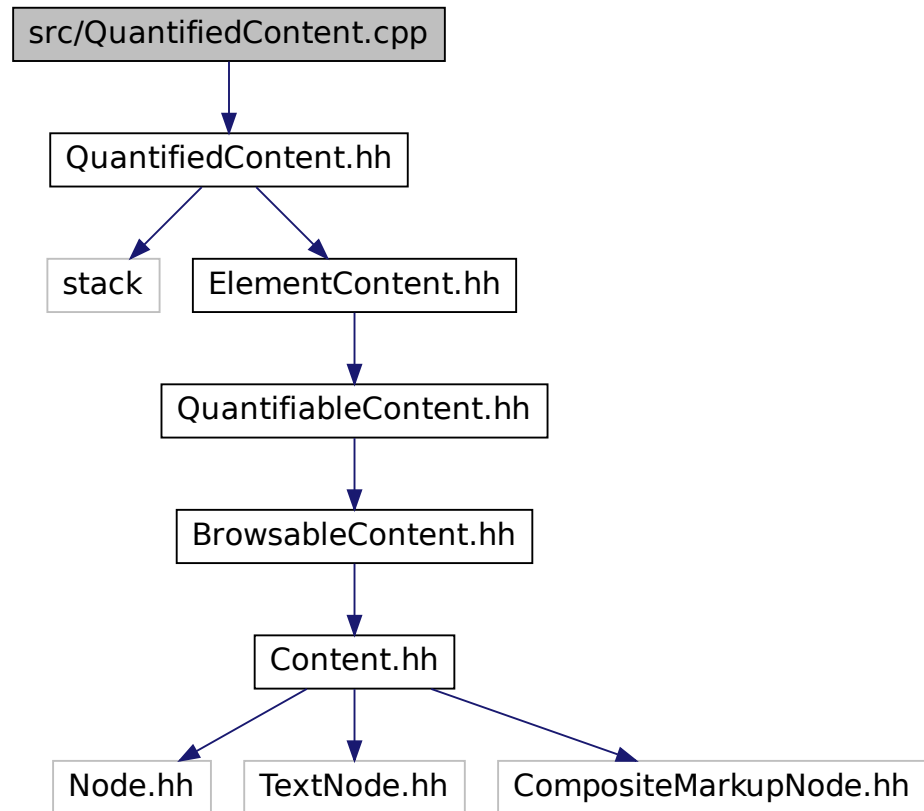
## Espaces de nommage

- namespace `dtd`

## 7.29 Référence du fichier src/QuantifiedContent.cpp

```
#include "QuantifiedContent.hh"
```

Graphe des dépendances par inclusion de QuantifiedContent.cpp :



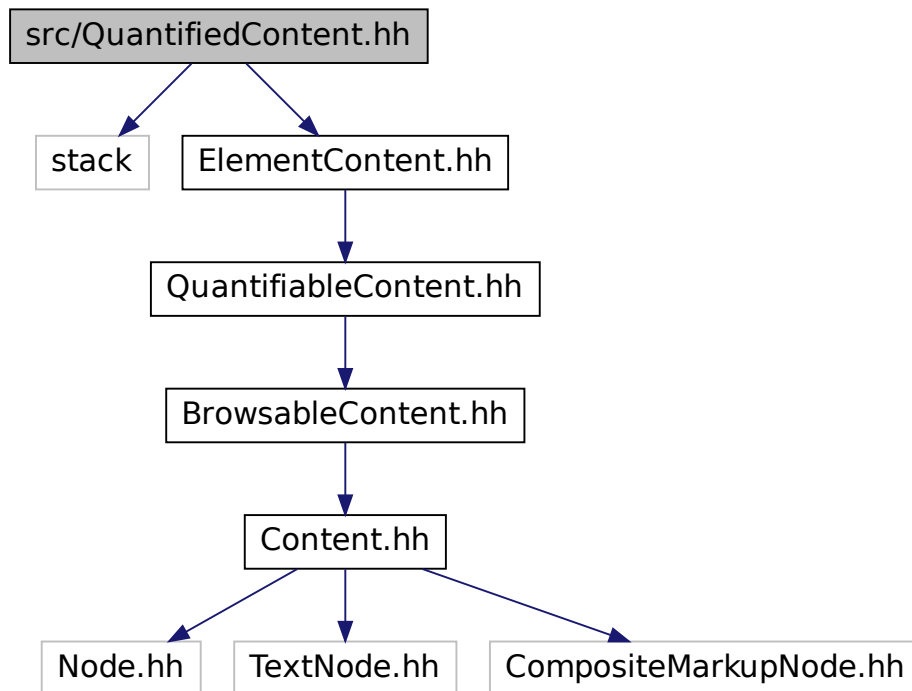
### Espaces de nommage

– namespace `dtd`

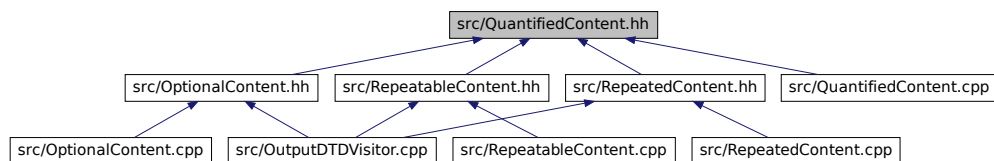
## 7.30 Référence du fichier `src/QuantifiedContent.hh`

```
#include <stack>
#include "ElementContent.hh"
```

Graphe des dépendances par inclusion de QuantifiedContent.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

– class `dtd` : `QuantifiedContent`

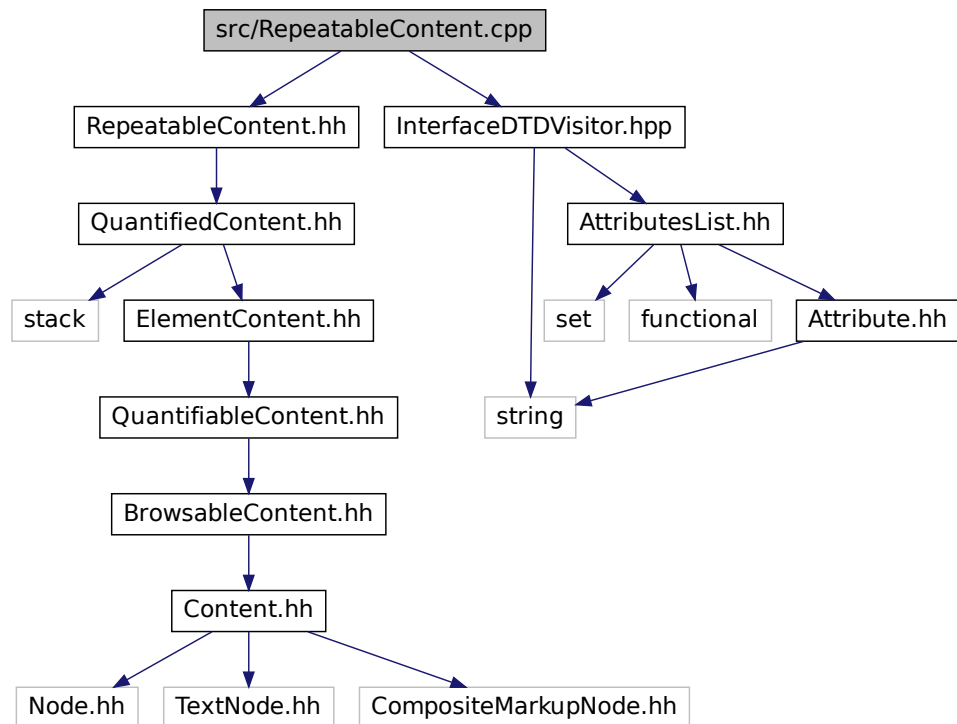
## Espaces de nommage

– namespace `dtd`

### 7.31 Référence du fichier src/RepeatableContent.cpp

```
#include "RepeatableContent.hh"
#include "InterfaceDTDVisitor.hpp"
```

Graphe des dépendances par inclusion de RepeatableContent.cpp :



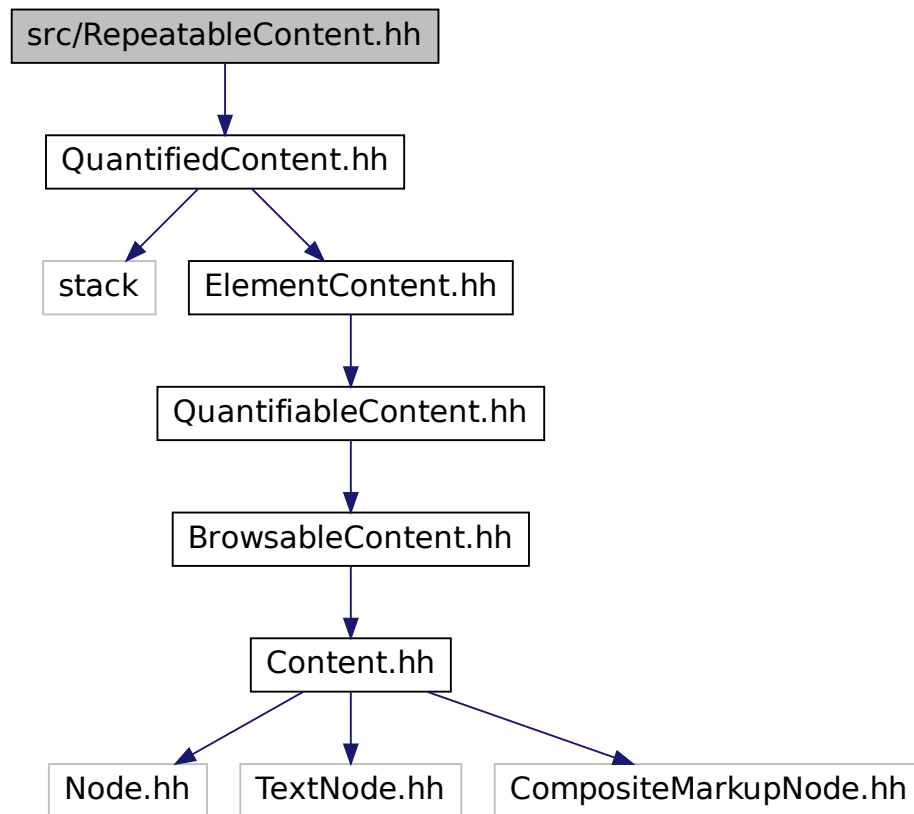
#### Espaces de nommage

– namespace `dtd`

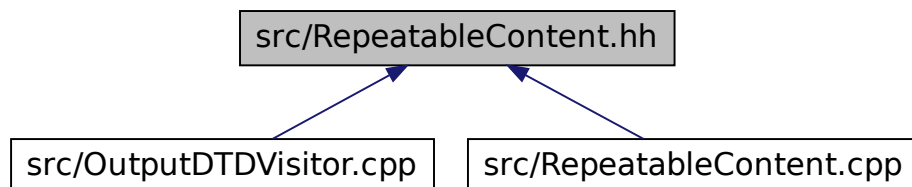
### 7.32 Référence du fichier src/RepeatableContent.hh

```
#include "QuantifiedContent.hh"
```

Graphe des dépendances par inclusion de RepeatableContent.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

- class `dtd : RepeatableContent`

## Espaces de nommage

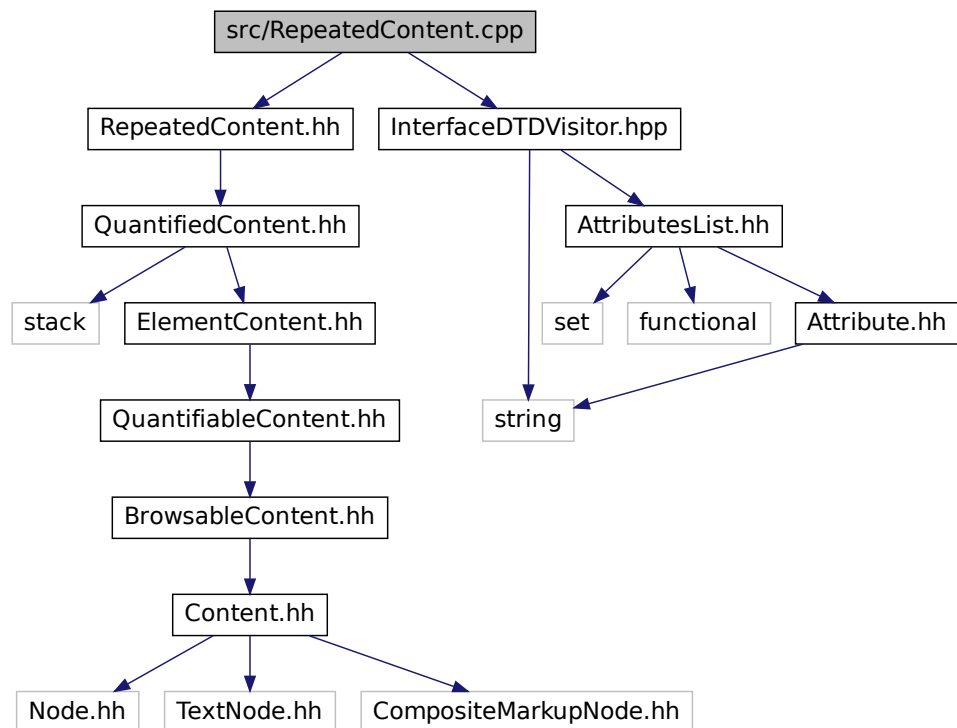
- namespace `dtd`

### 7.33 Référence du fichier `src/RepeatedContent.cpp`

```
#include "RepeatedContent.hh"
```

```
#include "InterfaceDTDVisitor.hpp"
```

Graphes des dépendances par inclusion de `RepeatedContent.cpp` :



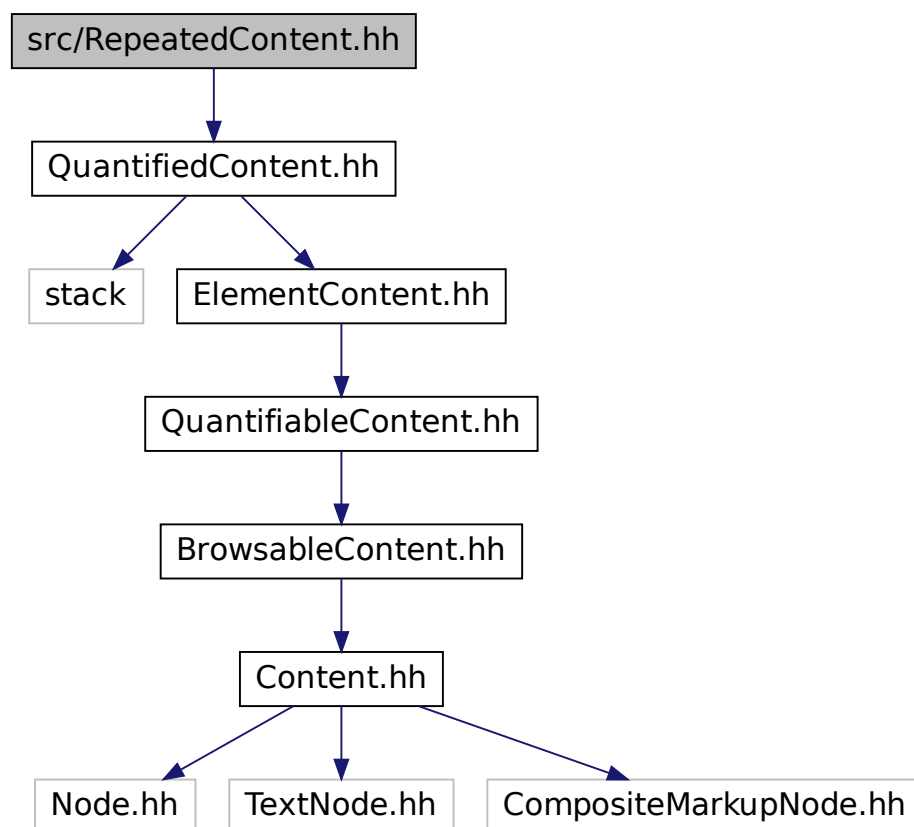
## Espaces de nommage

- namespace `dtd`

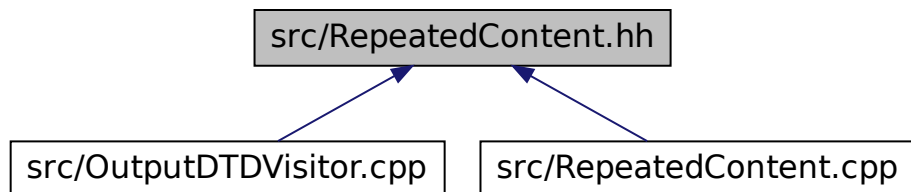
## 7.34 Référence du fichier src/RepeatedContent.hh

```
#include "QuantifiedContent.hh"
```

Graphe des dépendances par inclusion de RepeatedContent.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

- class `dtd : :RepeatedContent`

## Espaces de nommage

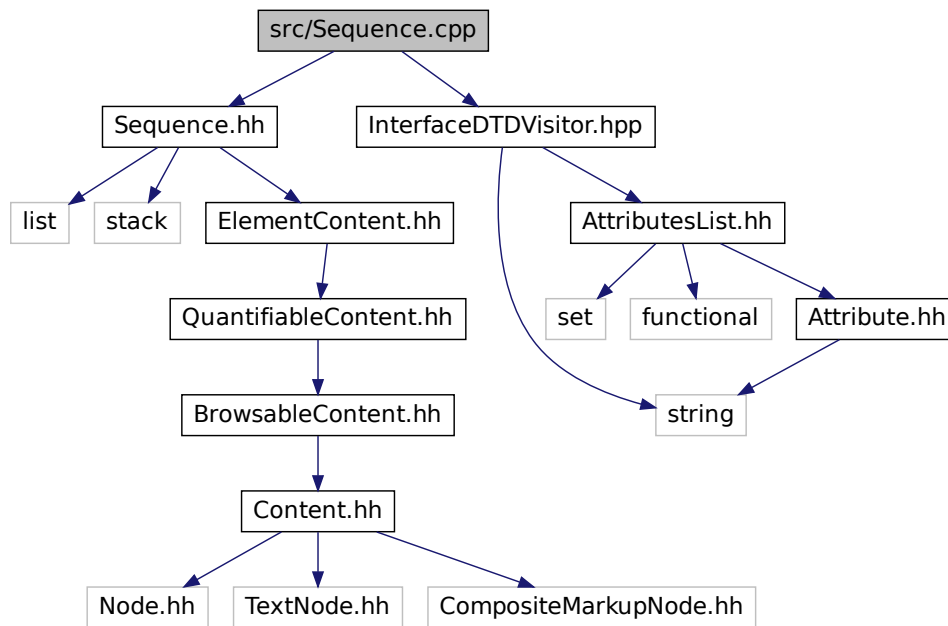
- namespace `dtd`

## 7.35 Référence du fichier `src/Sequence.cpp`

```
#include "Sequence.hh"  
#include "InterfaceDTDVisitor.hpp"
```



Grphe des dépendances par inclusion de Sequence.cpp :



## Espaces de nommage

– namespace `dtd`

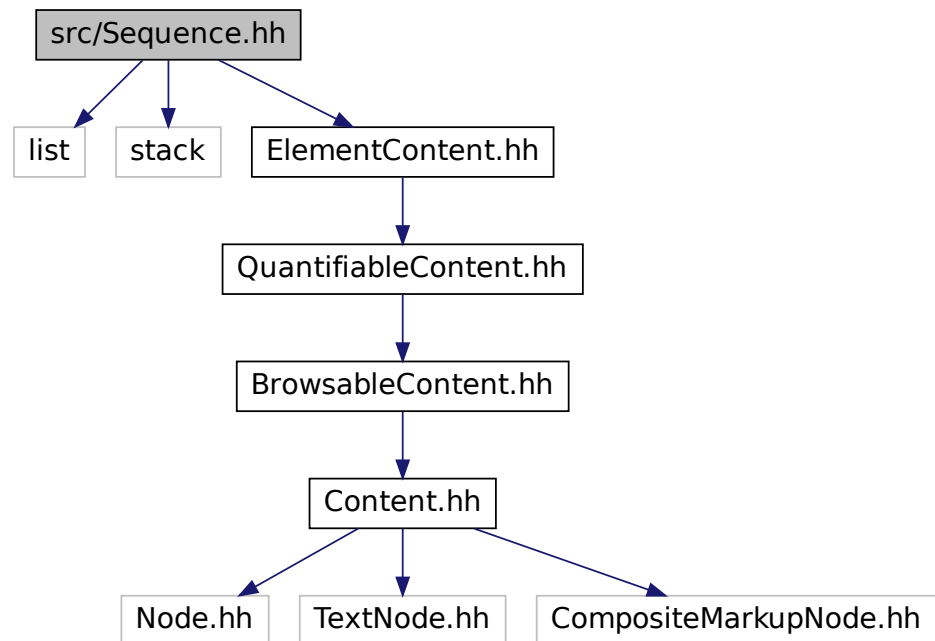
## 7.36 Référence du fichier src/Sequence.hh

```

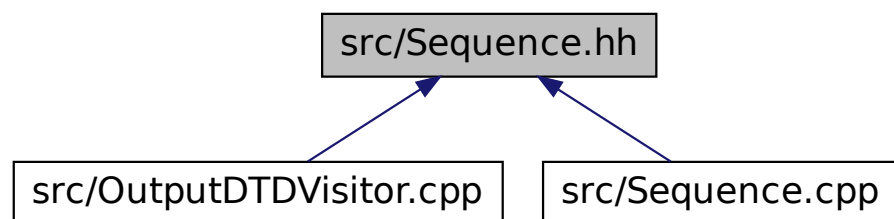
#include <list>
#include <stack>
#include "ElementContent.hh"

```

Graphe des dépendances par inclusion de Sequence.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

- class `dtd : :Sequence`
- struct `dtd : :Sequence : :_State`

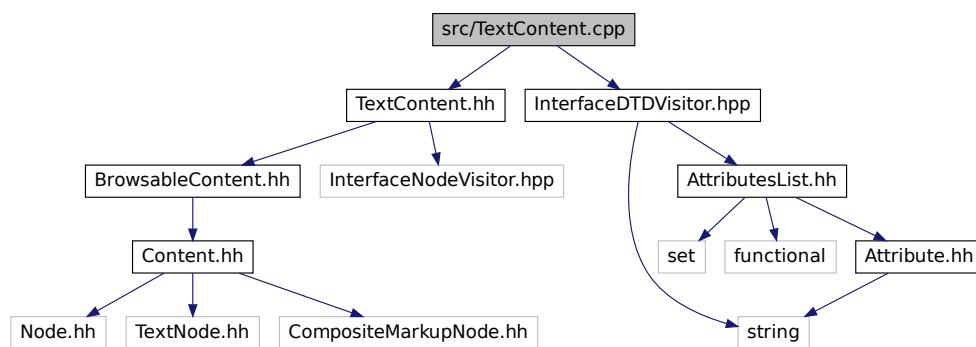
## Espaces de nommage

– namespace `dtd`

## 7.37 Référence du fichier src/TextContent.cpp

```
#include "TextContent.hh"
#include "InterfaceDTDVisitor.hpp"
```

Graphe des dépendances par inclusion de TextContent.cpp :



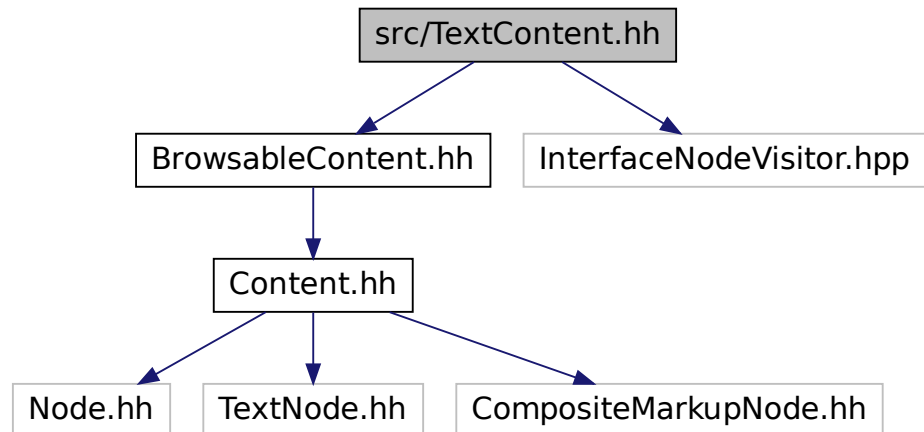
## Espaces de nommage

– namespace `dtd`

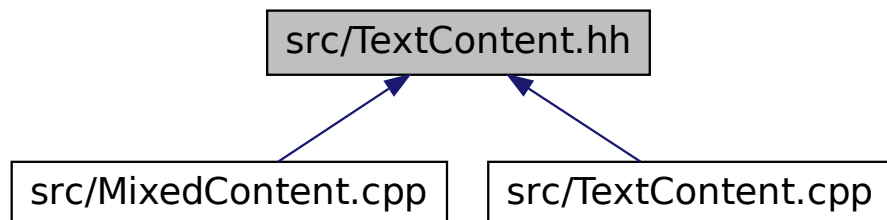
## 7.38 Référence du fichier src/TextContent.hh

```
#include "BrowsableContent.hh"
#include "InterfaceNodeVisitor.hpp"
```

Graphe des dépendances par inclusion de TextContent.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

– class `dtd :TextContent`

### Espaces de nommage

– namespace `dtd`

# Index

- ~AnyContent
  - dtd : :AnyContent, [23](#)
- ~Attribute
  - dtd : :Attribute, [24](#)
- ~BrowsableContent
  - dtd : :BrowsableContent, [28](#)
- ~Choice
  - dtd : :Choice, [34](#)
- ~Content
  - dtd : :Content, [38](#)
- ~DTD
  - dtd : :DTD, [40](#)
- ~ElementContent
  - dtd : :ElementContent, [46](#)
- ~ElementReference
  - dtd : :ElementReference, [49](#)
- ~EmptyContent
  - dtd : :EmptyContent, [54](#)
- ~InterfaceDTDVisitor
  - dtd : :InterfaceDTDVisitor, [57](#)
- ~MixedContent
  - dtd : :MixedContent, [65](#)
- ~OptionalContent
  - dtd : :OptionalContent, [70](#)
- ~OutputDTDVisitor
  - dtd : :OutputDTDVisitor, [75](#)
- ~QuantifiableContent
  - dtd : :QuantifiableContent, [83](#)
- ~QuantifiedContent
  - dtd : :QuantifiedContent, [86](#)
- ~RepeatableContent
  - dtd : :RepeatableContent, [91](#)
- ~RepeatedContent
  - dtd : :RepeatedContent, [95](#)
- ~Sequence
  - dtd : :Sequence, [100](#)
- ~TextContent
  - dtd : :TextContent, [105](#)
- \_AttributesLists
  - dtd : :DTD, [40](#)
- \_ChoosableSet
  - dtd : :Choice, [34](#)
- \_ElementId
  - dtd : :DTD, [40](#)
- \_Elements
  - dtd : :DTD, [40](#)
- \_OrderedContent
  - dtd : :Sequence, [100](#)
- \_State
  - dtd : :Choice : :\_State, [14](#)
  - dtd : :QuantifiableContent : :\_State, [17](#)
  - dtd : :Sequence : :\_State, [20](#)
- \_StatesStack
  - dtd : :Choice, [34](#)
  - dtd : :MixedContent, [64](#)
  - dtd : :QuantifiedContent, [86](#)
  - dtd : :Sequence, [100](#)
- \_afterValidation
  - dtd : :BrowsableContent, [28](#)
  - dtd : :Choice, [34](#)
  - dtd : :MixedContent, [65](#)
  - dtd : :QuantifiedContent, [87](#)
  - dtd : :Sequence, [100](#)
- \_attributesLists
  - dtd : :DTD, [42](#)
- \_beforeValidation
  - dtd : :BrowsableContent, [28](#)
  - dtd : :Choice, [34](#)
  - dtd : :MixedContent, [65](#)
  - dtd : :QuantifiedContent, [87](#)
  - dtd : :Sequence, [100](#)
- \_browseDown
  - dtd : :BrowsableContent, [29](#)
- \_browseUp
  - dtd : :BrowsableContent, [29](#)
- \_choice
  - dtd : :MixedContent, [67](#)
- \_choosable
  - dtd : :Choice, [36](#)
- \_content
  - dtd : :QuantifiedContent, [87](#)

- `_continueValidation`
  - `dtd : :BrowsableContent`, [29](#)
  - `dtd : :Choice`, [34](#)
  - `dtd : :MixedContent`, [65](#)
  - `dtd : :OptionalContent`, [70](#)
  - `dtd : :RepeatableContent`, [91](#)
  - `dtd : :RepeatedContent`, [95](#)
  - `dtd : :Sequence`, [100](#)
- `_dtd`
  - `dtd : :ElementReference`, [50](#)
- `_elements`
  - `dtd : :DTD`, [42](#)
- `_embeddedContent`
  - `dtd : :Sequence`, [102](#)
- `_indent`
  - `dtd : :OutputDTDVisitor`, [77](#)
- `_indentChar`
  - `dtd : :OutputDTDVisitor`, [77](#)
- `_indentUnit`
  - `dtd : :OutputDTDVisitor`, [77](#)
- `_isValid`
  - `dtd : :DTD`, [40](#)
- `_lastNodesValid`
  - `dtd : :DTD`, [42](#)
- `_matchResult`
  - `dtd : :ElementReference`, [50](#)
- `_name`
  - `dtd : :Attribute`, [25](#)
  - `dtd : :ElementReference`, [51](#)
- `_namespace`
  - `dtd : :ElementReference`, [51](#)
- `_newValidation`
  - `dtd : :BrowsableContent`, [30](#)
- `_out`
  - `dtd : :OutputDTDVisitor`, [77](#)
- `_stack`
  - `dtd : :Choice`, [36](#)
  - `dtd : :MixedContent`, [67](#)
  - `dtd : :QuantifiedContent`, [87](#)
  - `dtd : :Sequence`, [102](#)
- `_startValidation`
  - `dtd : :BrowsableContent`, [30](#)
  - `dtd : :Choice`, [35](#)
  - `dtd : :ElementReference`, [49](#)
  - `dtd : :MixedContent`, [66](#)
  - `dtd : :OptionalContent`, [70](#)
  - `dtd : :RepeatableContent`, [91](#)
  - `dtd : :RepeatedContent`, [96](#)
  - `dtd : :Sequence`, [101](#)
  - `dtd : :TextContent`, [105](#)
- `_textContent`
  - `dtd : :MixedContent`, [67](#)
- `_validRootName`
  - `dtd : :DTD`, [43](#)
- `_validatingRoot`
  - `dtd : :DTD`, [42](#)
- `_validationResult`
  - `dtd : :TextContent`, [106](#)
- `accept`
  - `dtd : :AnyContent`, [23](#)
  - `dtd : :Choice`, [35](#)
  - `dtd : :Content`, [38](#)
  - `dtd : :DTD`, [40](#)
  - `dtd : :ElementReference`, [49](#)
  - `dtd : :EmptyContent`, [54](#)
  - `dtd : :MixedContent`, [66](#)
  - `dtd : :OptionalContent`, [71](#)
  - `dtd : :RepeatableContent`, [92](#)
  - `dtd : :RepeatedContent`, [96](#)
  - `dtd : :Sequence`, [101](#)
  - `dtd : :TextContent`, [105](#)
- `addAttributesList`
  - `dtd : :DTD`, [41](#)
- `addElement`
  - `dtd : :DTD`, [41](#)
- `ANY_CONTENT_STR`
  - `dtd : :OutputDTDVisitor`, [77](#)
- `AnyContent`
  - `dtd : :AnyContent`, [23](#)
  - `dtd : :InterfaceDTDVisitor`, [60](#)
- `ATTLIST_MARKUP_NAME_STR`
  - `dtd : :OutputDTDVisitor`, [77](#)
- `Attribute`
  - `dtd : :Attribute`, [24](#)
- `AttributesList`
  - `dtd`, [9](#)
- `begin`
  - `dtd : :Choice`, [35](#)
  - `dtd : :MixedContent`, [66](#)
  - `dtd : :Sequence`, [101](#)
- `BrowsableContent`
  - `dtd : :BrowsableContent`, [28](#)
- `checkAttributes`
  - `dtd : :DTD`, [41](#)
- `Choice`
  - `dtd : :Choice`, [34](#)
  - `dtd : :InterfaceDTDVisitor`, [60](#)

- CHOICE\_SEPARATOR\_STR
  - dtd : :OutputDTDVisitor, 78
- ChoosableSet
  - dtd : :Choice, 34
  - dtd : :MixedContent, 64
- CLOSE\_MARKUP\_STR
  - dtd : :OutputDTDVisitor, 78
- const\_iterator
  - dtd : :Choice, 34
  - dtd : :MixedContent, 65
  - dtd : :Sequence, 100
- Content
  - dtd : :Content, 38
- content
  - dtd : :QuantifiedContent, 87
- DEFAULT\_INDENT\_CHAR
  - dtd : :OutputDTDVisitor, 78
- DEFAULT\_INDENT\_UNIT
  - dtd : :OutputDTDVisitor, 78
- doIndent
  - dtd : :OutputDTDVisitor, 75
- DTD
  - dtd : :DTD, 40
- dtd, 9
  - AttributesList, 9
- dtd : :AnyContent, 20
  - ~AnyContent, 23
  - accept, 23
  - AnyContent, 23
  - validate, 23
- dtd : :Attribute, 24
  - ~Attribute, 24
  - \_name, 25
  - Attribute, 24
  - name, 24
- dtd : :AttributesComparator, 25
  - operator(), 25
- dtd : :BrowsableContent, 25
  - ~BrowsableContent, 28
  - \_afterValidation, 28
  - \_beforeValidation, 28
  - \_browseDown, 29
  - \_browseUp, 29
  - \_continueValidation, 29
  - \_newValidation, 30
  - \_startValidation, 30
  - BrowsableContent, 28
  - validate, 30
- dtd : :Choice, 30
  - ~Choice, 34
  - \_ChoosableSet, 34
  - \_StatesStack, 34
  - \_afterValidation, 34
  - \_beforeValidation, 34
  - \_choosable, 36
  - \_continueValidation, 34
  - \_stack, 36
  - \_startValidation, 35
  - accept, 35
  - begin, 35
  - Choice, 34
  - ChoosableSet, 34
  - const\_iterator, 34
  - end, 35
  - validate, 35
- dtd : :Choice : : \_State, 11
  - \_State, 14
  - nextChoosable, 14
- dtd : :Content, 36
  - ~Content, 38
  - accept, 38
  - Content, 38
  - validate, 38
- dtd : :DTD, 39
  - ~DTD, 40
  - \_AttributesLists, 40
  - \_ElementId, 40
  - \_Elements, 40
  - \_attributesLists, 42
  - \_elements, 42
  - \_isValid, 40
  - \_lastNodesValid, 42
  - \_validRootName, 43
  - \_validatingRoot, 42
  - accept, 40
  - addAttributesList, 41
  - addElement, 41
  - checkAttributes, 41
  - DTD, 40
  - getAttributesList, 41
  - getElement, 42
  - isValid, 42
  - visit, 42
- dtd : :ElementContent, 43
  - ~ElementContent, 46
  - ElementContent, 46
- dtd : :ElementReference, 46
  - ~ElementReference, 49
  - \_dtd, 50

- [\\_matchResult](#), 50
  - [\\_name](#), 51
  - [\\_namespace](#), 51
  - [\\_startValidation](#), 49
  - [accept](#), 49
  - [ElementReference](#), 49
  - [matches](#), 50
  - [name](#), 50
  - [ns](#), 50
  - [visit](#), 50
- [dtd : :EmptyContent](#), 51
  - [~EmptyContent](#), 54
  - [accept](#), 54
  - [EmptyContent](#), 54
  - [validate](#), 54
- [dtd : :InterfaceDTDVisitor](#), 55
  - [~InterfaceDTDVisitor](#), 57
  - [AnyContent](#), 60
  - [Choice](#), 60
  - [ElementReference](#), 60
  - [EmptyContent](#), 60
  - [MixedContent](#), 60
  - [OptionalContent](#), 60
  - [RepeatableContent](#), 60
  - [RepeatedContent](#), 60
  - [Sequence](#), 61
  - [TextContent](#), 61
  - [visit](#), 57–59
  - [visitAttributesList](#), 59
  - [visitElement](#), 59
- [dtd : :MixedContent](#), 61
  - [~MixedContent](#), 65
  - [\\_StatesStack](#), 64
  - [\\_afterValidation](#), 65
  - [\\_beforeValidation](#), 65
  - [\\_choice](#), 67
  - [\\_continueValidation](#), 65
  - [\\_stack](#), 67
  - [\\_startValidation](#), 66
  - [\\_textContent](#), 67
  - [accept](#), 66
  - [begin](#), 66
  - [ChoosableSet](#), 64
  - [const\\_iterator](#), 65
  - [end](#), 66
  - [MixedContent](#), 65
  - [validate](#), 66
- [dtd : :OptionalContent](#), 67
  - [~OptionalContent](#), 70
  - [\\_continueValidation](#), 70
  - [\\_startValidation](#), 70
  - [accept](#), 71
  - [OptionalContent](#), 70
- [dtd : :OutputDTDVisitor](#), 71
  - [~OutputDTDVisitor](#), 75
  - [\\_indent](#), 77
  - [\\_indentChar](#), 77
  - [\\_indentUnit](#), 77
  - [\\_out](#), 77
  - [ANY\\_CONTENT\\_STR](#), 77
  - [ATTLIST\\_MARKUP\\_NAME\\_STR](#), 77
  - [CHOICE\\_SEPARATOR\\_STR](#), 78
  - [CLOSE\\_MARKUP\\_STR](#), 78
  - [DEFAULT\\_INDENT\\_CHAR](#), 78
  - [DEFAULT\\_INDENT\\_UNIT](#), 78
  - [doIndent](#), 75
  - [ELEMENT\\_MARKUP\\_NAME\\_STR](#), 78
  - [EMPTY\\_CONTENT\\_STR](#), 78
  - [END\\_CHOICE\\_STR](#), 78
  - [END\\_MIXED\\_STR](#), 78
  - [END\\_SEQUENCE\\_STR](#), 78
  - [INSIDE\\_MARKUP\\_SPACE\\_STR](#), 79
  - [MIXED\\_SEPARATOR\\_STR](#), 79
  - [OPEN\\_MARKUP\\_STR](#), 79
  - [OPTIONAL\\_QUANTIFIER\\_STR](#), 79
  - [OutputDTDVisitor](#), 75
  - [REPEATABLE\\_QUANTIFIER\\_STR](#), 79
  - [REPEATED\\_QUANTIFIER\\_STR](#), 79
  - [SEQUENCE\\_SEPARATOR\\_STR](#), 79
  - [START\\_CHOICE\\_STR](#), 79
  - [START\\_MIXED\\_STR](#), 79
  - [START\\_SEQUENCE\\_STR](#), 80
  - [TEXT\\_CONTENT\\_STR](#), 80
  - [visit](#), 75, 76
  - [visitAttributesList](#), 76
  - [visitElement](#), 77
- [dtd : :QuantifiableContent](#), 80
  - [~QuantifiableContent](#), 83
  - [QuantifiableContent](#), 83
- [dtd : :QuantifiableContent : :\\_State](#), 14
  - [\\_State](#), 17
  - [endToken](#), 17
  - [firstToken](#), 17
  - [nextStep](#), 17
- [dtd : :QuantifiedContent](#), 83
  - [~QuantifiedContent](#), 86
  - [\\_StatesStack](#), 86
  - [\\_afterValidation](#), 87
  - [\\_beforeValidation](#), 87
  - [\\_content](#), 87



- [\\_stack](#), [87](#)
  - [content](#), [87](#)
  - [QuantifiedContent](#), [86](#)
  - [validate](#), [87](#)
- [dtd : :RepeatableContent](#), [88](#)
  - [~RepeatableContent](#), [91](#)
  - [\\_continueValidation](#), [91](#)
  - [\\_startValidation](#), [91](#)
  - [accept](#), [92](#)
  - [RepeatableContent](#), [91](#)
- [dtd : :RepeatedContent](#), [92](#)
  - [~RepeatedContent](#), [95](#)
  - [\\_continueValidation](#), [95](#)
  - [\\_startValidation](#), [96](#)
  - [accept](#), [96](#)
  - [RepeatedContent](#), [95](#)
- [dtd : :Sequence](#), [96](#)
  - [~Sequence](#), [100](#)
  - [\\_OrderedContent](#), [100](#)
  - [\\_StatesStack](#), [100](#)
  - [\\_afterValidation](#), [100](#)
  - [\\_beforeValidation](#), [100](#)
  - [\\_continueValidation](#), [100](#)
  - [\\_embeddedContent](#), [102](#)
  - [\\_stack](#), [102](#)
  - [\\_startValidation](#), [101](#)
  - [accept](#), [101](#)
  - [begin](#), [101](#)
  - [const\\_iterator](#), [100](#)
  - [end](#), [101](#)
  - [OrderedContent](#), [100](#)
  - [Sequence](#), [100](#)
  - [validate](#), [102](#)
- [dtd : :Sequence : :\\_State](#), [17](#)
  - [\\_State](#), [20](#)
  - [nextEmbeddedContent](#), [20](#)
- [dtd : :TextContent](#), [102](#)
  - [~TextContent](#), [105](#)
  - [\\_startValidation](#), [105](#)
  - [\\_validationResult](#), [106](#)
  - [accept](#), [105](#)
  - [TextContent](#), [105](#)
  - [visit](#), [106](#)
- [ELEMENT\\_MARKUP\\_NAME\\_STR](#)
  - [dtd : :OutputDTDVisitor](#), [78](#)
- [ElementContent](#)
  - [dtd : :ElementContent](#), [46](#)
- [ElementReference](#)
  - [dtd : :ElementReference](#), [49](#)
- [dtd : :InterfaceDTDVisitor](#), [60](#)
- [EMPTY\\_CONTENT\\_STR](#)
  - [dtd : :OutputDTDVisitor](#), [78](#)
- [EmptyContent](#)
  - [dtd : :EmptyContent](#), [54](#)
  - [dtd : :InterfaceDTDVisitor](#), [60](#)
- [end](#)
  - [dtd : :Choice](#), [35](#)
  - [dtd : :MixedContent](#), [66](#)
  - [dtd : :Sequence](#), [101](#)
- [END\\_CHOICE\\_STR](#)
  - [dtd : :OutputDTDVisitor](#), [78](#)
- [END\\_MIXED\\_STR](#)
  - [dtd : :OutputDTDVisitor](#), [78](#)
- [END\\_SEQUENCE\\_STR](#)
  - [dtd : :OutputDTDVisitor](#), [78](#)
- [endToken](#)
  - [dtd : :QuantifiableContent : :\\_State](#), [17](#)
- [firstToken](#)
  - [dtd : :QuantifiableContent : :\\_State](#), [17](#)
- [getAttributesList](#)
  - [dtd : :DTD](#), [41](#)
- [getElement](#)
  - [dtd : :DTD](#), [42](#)
- [INSIDE\\_MARKUP\\_SPACE\\_STR](#)
  - [dtd : :OutputDTDVisitor](#), [79](#)
- [isValid](#)
  - [dtd : :DTD](#), [42](#)
- [matches](#)
  - [dtd : :ElementReference](#), [50](#)
- [MIXED\\_SEPARATOR\\_STR](#)
  - [dtd : :OutputDTDVisitor](#), [79](#)
- [MixedContent](#)
  - [dtd : :InterfaceDTDVisitor](#), [60](#)
  - [dtd : :MixedContent](#), [65](#)
- [name](#)
  - [dtd : :Attribute](#), [24](#)
  - [dtd : :ElementReference](#), [50](#)
- [nextChoosable](#)
  - [dtd : :Choice : :\\_State](#), [14](#)
- [nextEmbeddedContent](#)
  - [dtd : :Sequence : :\\_State](#), [20](#)
- [nextStep](#)

- dtd : :QuantifiableContent : :\_State, 17
- ns
  - dtd : :ElementReference, 50
- OPEN\_MARKUP\_STR
  - dtd : :OutputDTDVisitor, 79
- operator()
  - dtd : :AttributesComparator, 25
- OPTIONAL\_QUANTIFIER\_STR
  - dtd : :OutputDTDVisitor, 79
- OptionalContent
  - dtd : :InterfaceDTDVisitor, 60
  - dtd : :OptionalContent, 70
- OrderedContent
  - dtd : :Sequence, 100
- OutputDTDVisitor
  - dtd : :OutputDTDVisitor, 75
- QuantifiableContent
  - dtd : :QuantifiableContent, 83
- QuantifiedContent
  - dtd : :QuantifiedContent, 86
- REPEATABLE\_QUANTIFIER\_STR
  - dtd : :OutputDTDVisitor, 79
- RepeatableContent
  - dtd : :InterfaceDTDVisitor, 60
  - dtd : :RepeatableContent, 91
- REPEATED\_QUANTIFIER\_STR
  - dtd : :OutputDTDVisitor, 79
- RepeatedContent
  - dtd : :InterfaceDTDVisitor, 60
  - dtd : :RepeatedContent, 95
- Sequence
  - dtd : :InterfaceDTDVisitor, 61
  - dtd : :Sequence, 100
- SEQUENCE\_SEPARATOR\_STR
  - dtd : :OutputDTDVisitor, 79
- src/AnyContent.cpp, 107
- src/AnyContent.hh, 107
- src/Attribute.cpp, 108
- src/Attribute.hh, 109
- src/AttributesList.hh, 110
- src/BrowsableContent.cpp, 112
- src/BrowsableContent.hh, 112
- src/Choice.cpp, 113
- src/Choice.hh, 114
- src/Content.cpp, 116
- src/Content.hh, 116
- src/DTD.cpp, 117
- src/DTD.hh, 118
- src/ElementContent.cpp, 119
- src/ElementContent.hh, 120
- src/ElementReference.cpp, 122
- src/ElementReference.hh, 122
- src/EmptyContent.cpp, 124
- src/EmptyContent.hh, 124
- src/InterfaceDTDVisitor.hpp, 125
- src/MixedContent.cpp, 126
- src/MixedContent.hh, 127
- src/OptionalContent.cpp, 129
- src/OptionalContent.hh, 130
- src/OutputDTDVisitor.cpp, 131
- src/OutputDTDVisitor.hh, 132
- src/QuantifiableContent.cpp, 134
- src/QuantifiableContent.hh, 134
- src/QuantifiedContent.cpp, 135
- src/QuantifiedContent.hh, 136
- src/RepeatableContent.cpp, 138
- src/RepeatableContent.hh, 138
- src/RepeatedContent.cpp, 140
- src/RepeatedContent.hh, 141
- src/Sequence.cpp, 142
- src/Sequence.hh, 143
- src/TextContent.cpp, 145
- src/TextContent.hh, 145
- START\_CHOICE\_STR
  - dtd : :OutputDTDVisitor, 79
- START\_MIXED\_STR
  - dtd : :OutputDTDVisitor, 79
- START\_SEQUENCE\_STR
  - dtd : :OutputDTDVisitor, 80
- TEXT\_CONTENT\_STR
  - dtd : :OutputDTDVisitor, 80
- TextContent
  - dtd : :InterfaceDTDVisitor, 61
  - dtd : :TextContent, 105
- validate
  - dtd : :AnyContent, 23
  - dtd : :BrowsableContent, 30
  - dtd : :Choice, 35
  - dtd : :Content, 38
  - dtd : :EmptyContent, 54
  - dtd : :MixedContent, 66
  - dtd : :QuantifiedContent, 87
  - dtd : :Sequence, 102

---

visit  
  dtd : :DTD, [42](#)  
  dtd : :ElementReference, [50](#)  
  dtd : :InterfaceDTDVisitor, [57–59](#)  
  dtd : :OutputDTDVisitor, [75, 76](#)  
  dtd : :TextContent, [106](#)  
visitAttributesList  
  dtd : :InterfaceDTDVisitor, [59](#)  
  dtd : :OutputDTDVisitor, [76](#)  
visitElement  
  dtd : :InterfaceDTDVisitor, [59](#)  
  dtd : :OutputDTDVisitor, [77](#)